

AD-A134 989

IMAGE UNDERSTANDING RESEARCH(U) UNIVERSITY OF SOUTHERN
CALIFORNIA LOS ANGELES IMAGE PROCESSING INST

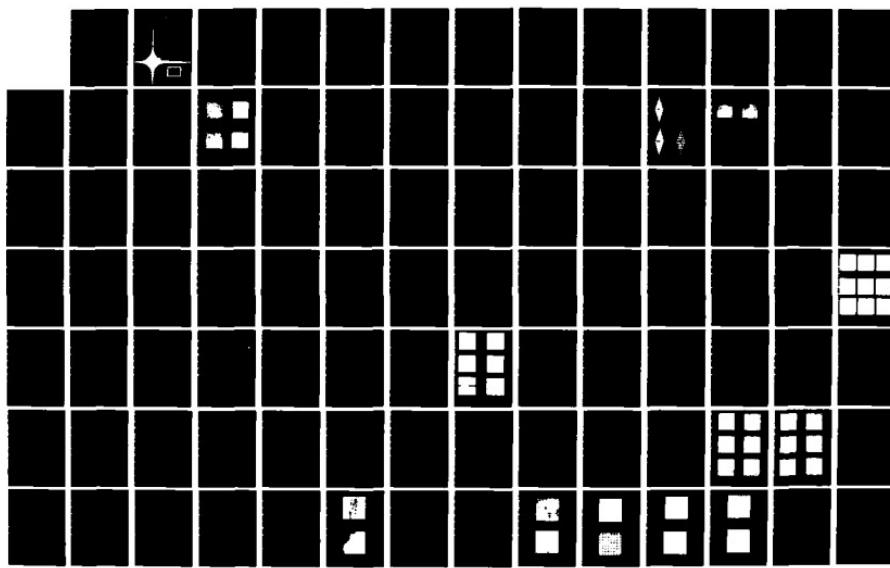
1/4

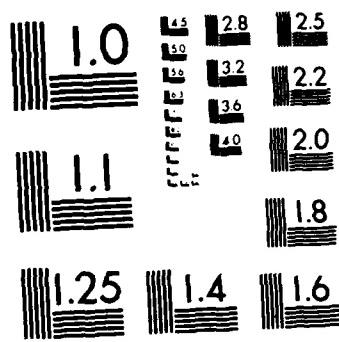
H C ANDREWS 30 SEP 78 USCIP1-840 F33615-76-C-1203

UNCLASSIFIED

F/G 20/6

NL



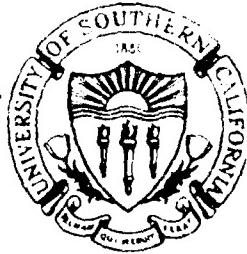


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

**E
N
G
I
N
E
R
I
N
G**

AD-A134989

USCIP Report 840



(1)

UNIVERSITY OF SOUTHERN CALIFORNIA

SEMIANNUAL TECHNICAL REPORT

Harry C. Andrews
Project Director

Covering Research Activity During the Period
1 April 1978 through 30 September 1978

30 September 1978

Image Processing Institute
University of Southern California
University Park
Los Angeles, California 90007

Sponsored by

Advanced Research Projects Agency
Contract No. F-33615-76-C-1203
ARPA Order No. 3119

DTC FILE COPY

DISSEMINATION STATEMENT A	
Approved for public release; Distribution Unlimited	



IMAGE PROCESSING INSTITUTE

83 11 25 020

SEMIANNUAL TECHNICAL REPORT

Covering Research Activity During the Period
1 April 1978 through 30 September 1978

Harry C. Andrews
Project Director
(213) 741-5514

Image Processing Institute
University of Southern California
University Park
Los Angeles, California 90007

30 September 1978

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Avail and/or	
Dist	Special

A-1

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Wright Patterson Air Force Base under Contract F-33615-76-C-1203, ARPA Order No. 3119. Additional support was provided by AFOSR Contract AFOSR-77-3285 and WPAFB Contract F-33615-77-C-1016.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER USCIPI Report 840	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) IMAGE UNDERSTANDING RESEARCH		5. TYPE OF REPORT & PERIOD COVERED Semianual Technical 1 April 1977-30 September 1978
7. AUTHOR(s) Harry C. Andrews (Project Director)		6. PERFORMING ORG. REPORT NUMBER F-33615-76-C-1203
9. PERFORMING ORGANIZATION NAME AND ADDRESS Image Processing Institute University of Southern California University Park, Los Angeles, Calif. 90007		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS ARPA Order No. 3119
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE September 30, 1978
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Wright Patterson Air Force Base U.S. Air Force Air Force Avionics Laboratory Air Force Systems Command, Ohio 45433		13. NUMBER OF PAGES 293
15. SECURITY CLASS. (of this report) UNCLASSIFIED		
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE		
16. DISTRIBUTION STATEMENT (of this Report) Approved for release: distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Key Words: Digital Image Processing, Image Restoration, Degrees of Freedom, Scene Analysis, Image Understanding, Edge Detection, Image Segmentation, CCD Arrays, CCD Processors.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This technical report summarizes the image understanding, image processing, and smart sensor research activities performed by the USC Image Processing Institute during the period of 1 April 1978 through 30 September 1978 under contract number F-33615-76-C-1203 with the Advanced Research Projects Agency, Information Processing Techniques Office, and monitored by the Wright-Patterson Air Force Base, Dayton, Ohio.		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

The research program has, as its primary purpose, the development of techniques and systems for understanding images. Methodologies range from low level image processing principles, smart sensor CCD LSI circuit design, up to higher level symbolic representations and relational structure manipulations. Results reported herein include quantitative edge evaluation, texture analysis and synthesis, feature combinations for linking and line development as well as the formulation of a higher level software system to handle storage and relational structures. Results in the signal processing area include Poisson MAP restoration, blind complex OTF deconvolution, condition number calculation, and inverse SAR imaging. Smart Sensor design of new 7x7 circuitry is progressing and real time TV rate 3x3 CCD chips are now operational and available.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Abstract

This technical report summarizes the image understanding, image processing, and smart sensor research activities performed by the USC Image Processing Institute during the period of 1 April 1978 through 30 September 1978 under contract number F-33615-76-C-1203 with the Advanced Research Projects Agency, Information Processing Techniques Office, and monitored by the Wright-Patterson Air Force Base, Dayton, Ohio.

The research program has, as its primary purpose, the development of techniques and systems for understanding images. Methodologies range from low level image processing principles, smart sensor CCD/LSI circuit design, up to higher level symbolic representations and relational structure manipulations. Results reported herein include quantitative edge evaluation, texture analysis and synthesis, feature combinations for linking and line development as well as the formulation of a higher level software system to handle storage and relational structures. Results in the signal processing area include Poisson MAP restoration, blind complex OTF deconvolution, condition number calculation, and inverse SAR imaging. Smart Sensor design of new 7x7 circuitry is progressing and real time TV rate 3x3 CCD chips are now operational and available.

PROJECT PARTICIPANTS

Project Director

Harry C. Andrews

Affiliation

Computer Science & Electrical
Engineering

Research Staff

Nasser E. Nahi

Affiliation

Electrical Engineering

Ramakant Nevatia

Computer Science

William K. Pratt

Electrical Engineering

Keith E. Price

Image Processing Institute

Alexander A. Sawchuk

Electrical Engineering

Timothy C. Strand

Image Processing Institute

Support Staff

Marilyn Chan

Ray Schmidt

Gary Edwards

James Tertocha

John Horner

Roger Tertocha

Eileen Jurak

Thomas Tertocha

Hilda Marti

Rose Ward

Toyone Mayeda

Steven White

Charles McManis

Amy Yiu

Clay Olmstead

Students

Ikram E. Abdou	David Drake
Ahmad Armand	David Garber
Behnam Ashjari	Charles F. Hall
K. Ramesh Babu	Chung-Kai Hsueh
Pradeep Bhadsavle	Kenneth I. Laws
Bir Bhanu	Sang Uk Lee
Chung-Ching Chen	Chun Moo Lo
Peter Chuan	John Morton

TABLE OF CONTENTS

	Page
1. Research Overview.....	1
2. Image Understanding Projects.....	5
2.1 Stochastic Based Visual Texture Feature Extraction	
- William K. Pratt and Olivier D. Faugeras....	6
2.2 Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors	
- Ikram E. Abdou and William K. Pratt.....	28
2.3 Optical Pseudocolor Encoding of One-Dimensional Texture Patterns	
- Timothy C. Strand and David D. Garber.....	46
2.4 One-Dimensional Texture Pattern Generation and Discrimination	
- David D. Garber.....	54
2.5 Experiments in Natural Texture Description	
- Keith E. Price and Ramakant Nevatia.....	77
2.6 Representation and Acquisition of High-Level Image Descriptions	
- Keith E. Price.....	89
2.7 A Proposed Class of Picture Operators	
- Kenneth I. Laws.....	96
2.8 An Edge Detection, Linking and Line Finding Program	
- Ramakant Nevatia and K. Ramesh Babu.....	103
2.9 Descriptions of Linear Segment Objects	
- K. Ramesh Babu and Ramakant Nevatia.....	116
3. Image Processing Projects.....	122
3.1 Condition Number Computation of a Discrete Deconvolution Operator	

	Page
	- Ikram E. Abdou and William K. Pratt..... 123
3.2	Estimation of Image Signal with Poisson Noise - I
	- Chun Moo Lo and Alexander A. Sawchuk..... 135
3.3	Computer Hologram Interpolation with the DFT
	- Chung-Kai Hsueh and Alexander A. Sawchuk.... 156
3.4	Spotlight S.A.R. Imaging Using RAT-SCAT Site Date
	- Peter Chuan..... 175
3.5	Blind OTF Restoration
	- John B. Morton and Harry C. Andrews..... 195
3.6	Target Motion Induced Radar Imaging
	- Chung-Ching Chen and Harry C. Andrews..... 212
4.	Smart Sensor Projects..... 247
4.1	Charge Coupled Device Image Processing Circuitry
	- Graham R. Nudd..... 248
5.	Recent Ph.D. Dissertations..... 279
5.1	Quantitative Methods of Edge Detection
	- Ikram E. Abdou..... 280
5.2	An Investigation Into an A Posteriori Method of Image Restoration
	- John B. Morton..... 280
5.3	Imaging With Radar Returns
	- Chung-Ching Chen..... 281
6.	Recent Institute Personnel Publications..... 283

1. Research Overview

This document represents the results of research developed over the past 6 months at the USC Image Processing Institute. Research has been devoted to 3 major areas: image understanding, image processing, and smart sensor design. These areas are abstracted below.

Image Understanding Projects

The image understanding tasks presented in this semiannual report are focused in first level and second or higher level processing procedures. In the first level processes, edge and texture techniques are developed. Edge analysis results are presented in which quantitative measures of performance on a variety of different edge operators are evaluated. Different performance functions, such as edge detection, positional accuracy, invariance of operator to orientation, etc., are utilized. In the area of texture work both analysis and synthesis procedures are reported. Texture analysis via optical filtering and the use of color representation has been demonstrated to be an effective means of detection and visualization of specific texture patterns. In the synthesis of texture a stochastic whitening process is developed which looks extremely hopeful as a tool in defining features for texture recognition and discrimination. Another texture synthesis technique is presented which is based upon the statistical (N-gram) approach. This method although still in its one-dimensional form, show promise in its avoidance of moment techniques. Finally, some novel "segmented window" first layer processing techniques are presented with hypotheses as to their usefulness in ongoing research.

In the arena of second or higher level processors, feature usages of small Fourier transforms on reflectance imagery, edges, direction of edges and density of edges is developed. Edge detection, linking, and line finding algorithms as well as descriptions of linear segmented objects are presented as work in progress for various image segmentation scenarios. Finally, higher level operating software principles are formulated and examples of data structures and their relationships are presented.

Image Processing Projects

A variety of image processing projects are reported herein. They fall into three general areas of computational procedures, restoration methodologies, and inverse SAR imaging. A presentation is made on the computation of the condition number of a matrix to predict the degree of ill-conditioning and subsequent potential degrees of freedom in such a process. Such computations become extremely useful for large matrix processes as found in most imaging applications. In the generation of computer hologram interpolations, a special computational savings is developed to avoid the inefficiencies of zero padding traditionally used in most Fourier image filtering techniques.

In the arena of image restoration two techniques are reported upon. Results from the method of blind a posteriori restoration are presented in pictorial form. A new method of Poisson MAP restoration is also developed and analysis presented in which improved sensor models for imaging result.

Finally, two papers on inverse synthetic aperture radar imaging are presented. One is formative in its presentation and proposes to image shadowed regions via RATSCAT turntable

data. The second represents processing results from an inflight aircraft in both a straight flight and a turn set of geometries. Resulting imagery is presented.

Smart Sensor Projects

The following report from Hughes Research Laboratories reflects the continuing progress on the CCD smart sensor design front. As usual we are pleased to see such results and wish to point out that this represents a classic illustration of technology transfer as the US Army NVL has contracted and received one of our earlier circuit chips in an operating unit. Recent chip design will afford 7x7 processing as well as programmable arrays and limited feature selection in our ultimate effort for the computation of a texture CCD circuit.

Recent Graduates

One of the Image Processing Institutes' most precious products is its graduate students and it is always a pleasure to see our students graduate and move on to professional positions. This section lists the abstracts of the dissertations of the three most recent graduates and represents research in edge detection, restoration, and radar imaging. We are proud of their work and wish them well in their endeavors. Details of their dissertations appear as USCIPI technical reports and are available upon request for those interested.

Recent Publications

The report closes with a listing of Institute research staff publications. The majority of these are in the reviewed open literature and are an indication of the health

of our research ideas. Naturally, due to the review process a delay in published results occurs for the open literature publications.

2. Image Understanding Project

The image understanding tasks presented in this semiannual report are focused in first level and second or higher level processing procedures. In the first level processes, edge and texture techniques are developed. Edge analysis results are presented in which quantitative measures of performance on a variety of different edge operators are evaluated. Different performance functions, such as edge detection, positional accuracy, invariance of operator to orientation, etc., are utilized. In the area of texture work both analysis and synthesis procedures are reported. Texture analysis via optical filtering and the use of color representation has been demonstrated to be an effective means of detection and visualization of specific texture patterns. In the synthesis of texture a stochastic whitening process is developed which looks extremely hopeful as a tool in defining features for texture recognition and discrimination. Another texture synthesis technique is presented which is based upon the statistical (N-gram) approach. This method although still in its one-dimensional form, show promise in its avoidance of moment techniques. Finally, some novel "segmented window" first layer processing techniques are presented with hypotheses -; to their usefulness in ongoing research.

In the arena of second or higher level processors, feature usages of small Fourier transforms on reflectance imagery, edges, direction of edges and density of edges is developed. Edge detection, linking, and line finding algorithms as well as descriptions of linear segmented objects are presented as work in progress for various image segmentation scenarios. Finally, higher level operating software principles are formulated and examples of data structures and their relationships are presented.

2.1 Stochastic-Based Visual Texture Feature Extraction

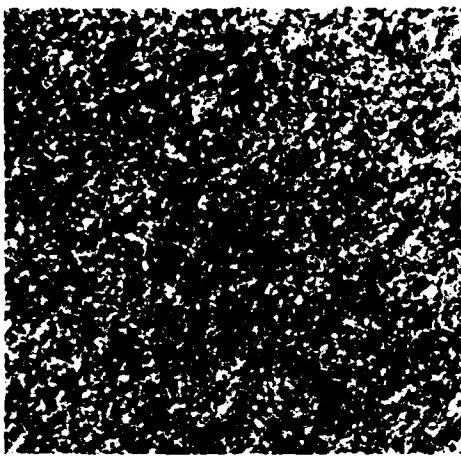
William K. Pratt and Olivier D. Faugeras*

Introduction

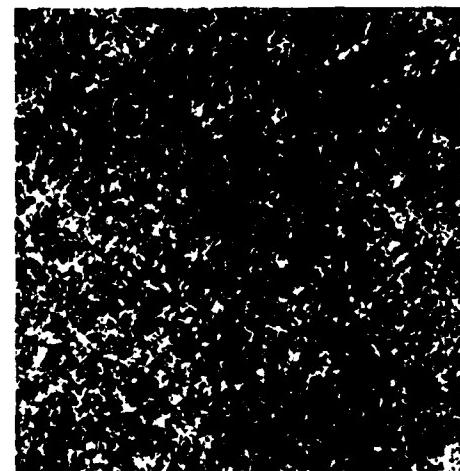
Image texture is a region property or feature of an image that characterizes the structural relationship of pixels within the region. The structural relationship of texture may be regarded from a deterministic or stochastic standpoint. In the deterministic formulation [1,2], texture is considered as a basic local pattern that is periodically or quasi-periodically repeated over some area. This definition is applicable to line patterns such as ruled line arrays, tiling patterns, etc. The stochastic formulation is based on a model in which a texture region is viewed as a sample of a two-dimensional stochastic process describable by its statistical parameters. This formulation is obviously applicable to the texture fields generated from random number arrays that have been so widely used in perceptual experiments [3-9]. In addition, the formulation seems well suited for natural textures consisting of isolated areas from multi-gray level images such as grass, water, forestry, etc. Figure 1 contains several examples of natural textures taken from the Brodatz [10] photographic album of natural texture.

The deterministic and stochastic definitions of texture that have been presented do not depend upon visual perception. The basic pattern and repetition frequency of a

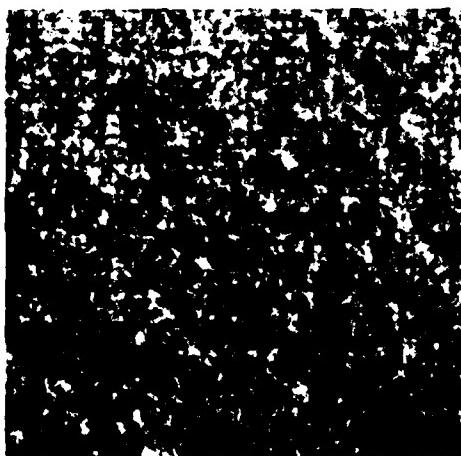
*Dr. O.D. Faugeras is with Institut de Recherche d'Informatique et d'Automatique, Domaine de Voluceau - Rocquencourt, 78150 Le Chesnay, France.



(a) sand



(b) grass



(c) wool



(d) raffia

Figure 1. Examples of Brodatz Texture Fields

texture sample could be perceptually invisible though quantitatively present. It may be desireable in some applications to characterize or classify such textures. However, the scope of this paper is limited to stochastic descriptions of visual texture that are in agreement with human perception.

Stochastic Texture Generation Experiments

Figure 2 contains a block diagram for a general model of stochastic texture generation. An array of independent, identically distributed random variables $W(j,k)$ passes through a linear or nonlinear spatial operator $\mathcal{O}\{.\}$ to produce a stochastic texture array $F(j,k)$. By controlling the form of the generating probability density $p(W)$ and the spatial operator $\mathcal{O}\{.\}$, it is possible to create texture fields with specified statistical properties.

Visual discrimination testing can be performed on the stochastic texture fields generated by the model of figure 2 to determine what statistical parameters are of perceptual importance. Examples of such testing by Pratt, Faugeras, and Gagalowicz [9] are summarized below:

1. Observers are sensitive to differences in first order densities of texture field pairs.
2. Observers are sensitive to differences in second order densities of correlated texture field pairs that possess the same first order densities.
3. Observers are not sensitive to differences in third order densities of uncorrelated or correlated texture field pairs that possess the same pairwise first and second order densities.
4. Observers are sensitive to differences in the autocorrelation of correlated texture field pairs

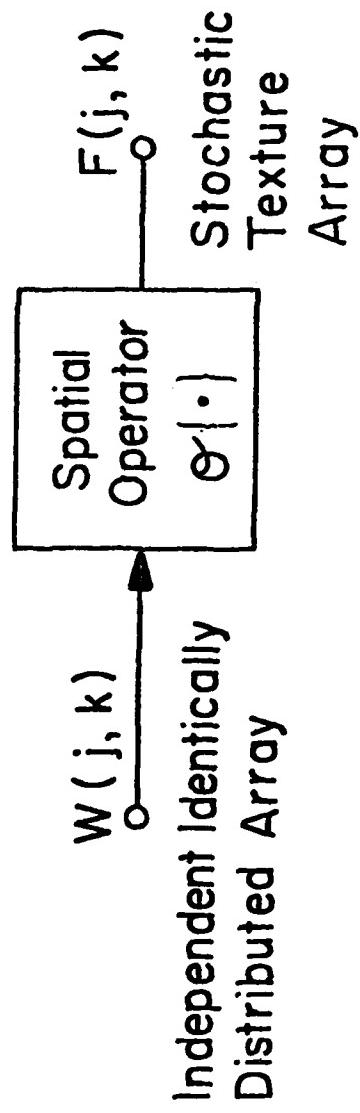


Figure 2. Stochastic texture field generation model

with common mean and variance.

5. Discriminable texture field pairs can be produced that possess the same mean, variance, and autocorrelation function.

These experimental conclusions establish useful bounds for developing stochastic-based visual texture features. Second order statistical measures should be sufficient, but the mean, variance, and autocorrelation function measures, by themselves, although directly or indirectly necessary, are not sufficient. The task then is to determine those stochastic features that are perceptually sufficient.

Stochastic-Based Texture Features

The most general second order stochastic description of a texture field is, of course, the second order joint density between all pixel pairs of a texture block. The joint density can be approximated by joint gray scale histograms of a texture block. The difficulty with this approach is its enormous dimensionality; N^2 two-dimensional histogram are produced for each $N \times N$ array of pixel neighbors and each histogram is an $L \times L$ array for L luminance levels. Thus, the raw feature contains on the order of N^2L^2 components! Clearly, some form of feature selection is necessary.

Haralick et. al. [11,12] have suggested texture features based on the spread of the joint histograms about their diagonals. If a textured region is highly correlated, the histograms values will tend to be concentrated toward the diagonal, while for an uncorrelated region, the histogram will tend toward uniformity. The Haralick family of texture features has proved to be effective in terms of classification accuracy [13] and stochastic analysis [14].

However, the feature extraction procedure suffers major handicaps: enormous computation is required for generation of the histograms and feature calculation; and there is an accuracy limitation in characterizing low contrast texture. There is a definite need to determine simpler texture features than those based on two-dimensional histograms.

Texture Field Decorrelation Techniques

From the stochastic texture generation model of figure 2, it is observed that fields generated by that model can be described quite compactly by specification of the spatial operator $\{\cdot\}$ and the stationary first order probability density $p(W)$ of the independent, identically distributed generating process $W(j,k)$. Such information cannot generally be determined from the texture field observation $F(j,k)$. However, this concept serves as a useful guide to the development of candidate texture features.

Consider the stationary ensemble autocorrelation function

$$K_F(m,n) = E\{F(j,k)F(j+m,k+n)\} \quad (1)$$

defined for lag values $m,n = 0, \pm 1, \pm 2, \dots, \pm T$ where $E\{\cdot\}$ denotes the expectation operator. The ensemble autocorrelation function can be estimated by the spatial autocorrelation function

$$A_F(m,n) = \sum_{u=j-W}^{j+W} \sum_{v=k-W}^{k+W} F(u,v)F(u-m,v-n) \quad (2)$$

where computation is over a $(2W+1) \times (2W+1)$ window. It is possible to perform a whitening transformation [15,p.556], based on the measured autocorrelation function of eq. (2), to produce an uncorrelated, identically distributed field

$$\hat{W}(j,k) = H(j,k) * H(j,k) \quad (3)$$

where $H(j,k)$ is the whitening operator and the symbol $*$ denotes convolution. By definition, the autocorrelation of the whitened image field is

$$\hat{K}_W(m,n) = \begin{cases} 1 & \text{if } m = n \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

The whitened field $\hat{W}(j,k)$ can be utilized as an estimate of the independent, identically distributed generating process $W(j,k)$.

If $W(j,k)$ were known exactly, then in principle, system identification techniques could be employed to estimate the spatial operator $\mathcal{O}\{\cdot\}$ from the texture observation $F(j,k)$. But, the whitened field estimate $\hat{W}(j,k)$ will only identify the spatial operator in terms of the autocorrelation function of $F(j,k)$, which is not unique. Thus, it is concluded that the probability density of the whitened field $p(\hat{W})$ and the spatial autocorrelation function of the texture field $K_F(m,n)$ are, in general, incomplete descriptors of the stochastic process $F(j,k)$. But, it may be possible that they are sufficient descriptors of its texture from the standpoint of visual texture discrimination.

Figure 3 provides examples of the measured spatial autocorrelation function of the natural texture fields of figure 1. Whitened fields corresponding to these texture fields are presented in figure 4 and first order amplitude histograms of the whitened fields are shown in Figure 5. Examination of the histograms indicates that they are all different. These experiments qualitatively support the contention that the spatial autocorrelation function of a texture field plus the first order amplitude histogram of its whitened texture field provide sufficient information for texture discrimination.

An obvious disadvantage of the whitening operator method of texture field decorrelation is the large amount of computation involved in the process. The experimental autocorrelation function of a texture block must first be formed, then the whitening operator must be generated, and finally the block must be processed. An alternative to this procedure is to utilize a gradient operator, such as a Laplacian or Sobel operator, that approximates the whitening operator. This topic is presently under investigation.

Feature Extraction

The previous sections have provided qualitative evidence that the autocorrelation of a texture field plus the first order amplitude histogram of its decorrelated field contain sufficient information for texture discrimination. Consideration must now be given to means of extracting this information and forming texture features useful for classification and analysis.

Figure 6 contains a block diagram of the stochastic-based feature extraction method. Quantitative techniques are now developed for representation of the

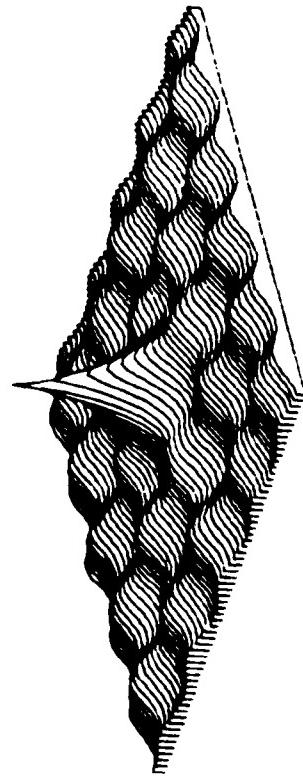
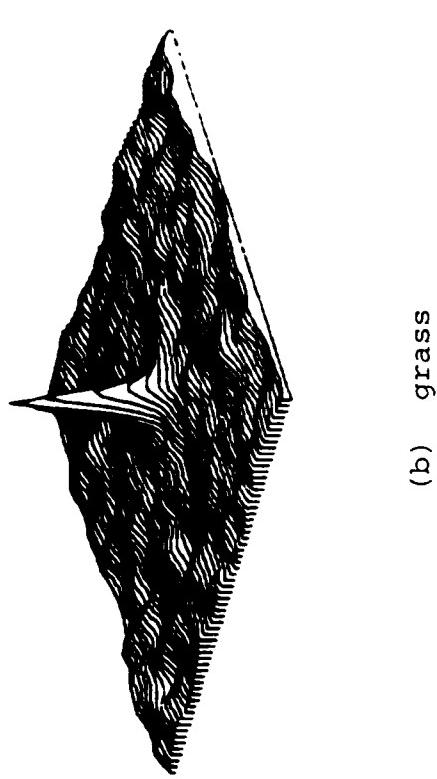
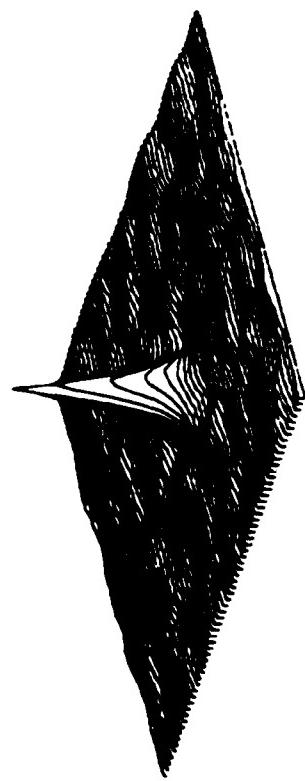
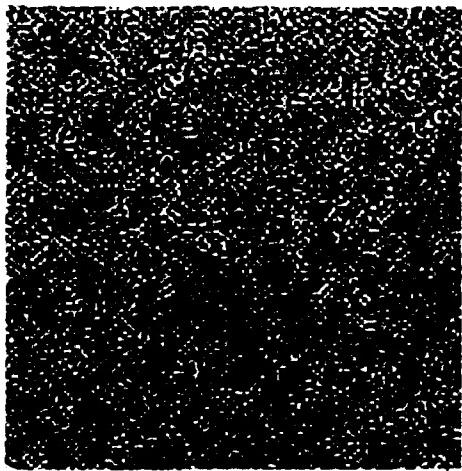
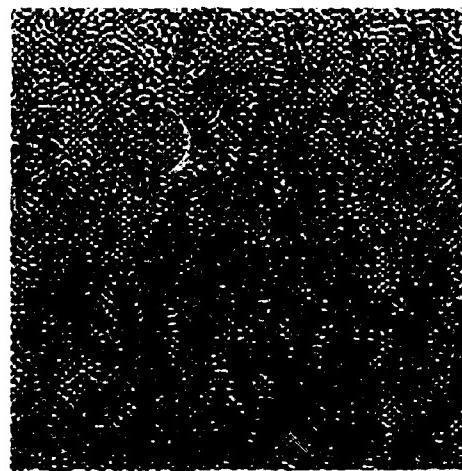


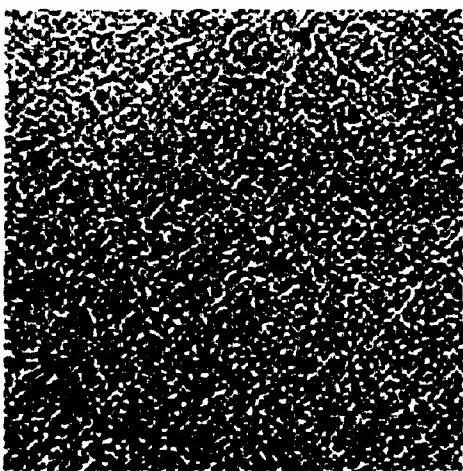
Figure 3. Perspective views of autocorrelation functions of natural texture fields
(c) wool
(d) raffia



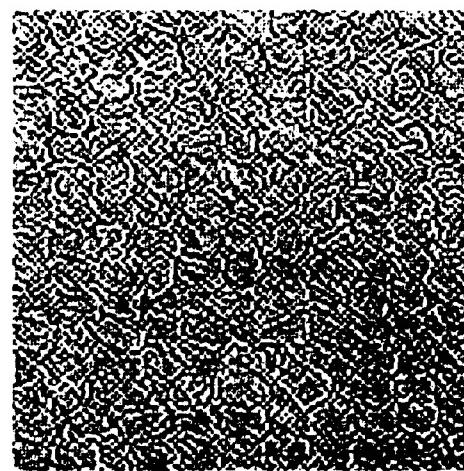
(a) sand



(b) grass



(c) wool



(d) raffia

Figure 4. Whitened natural texture fields

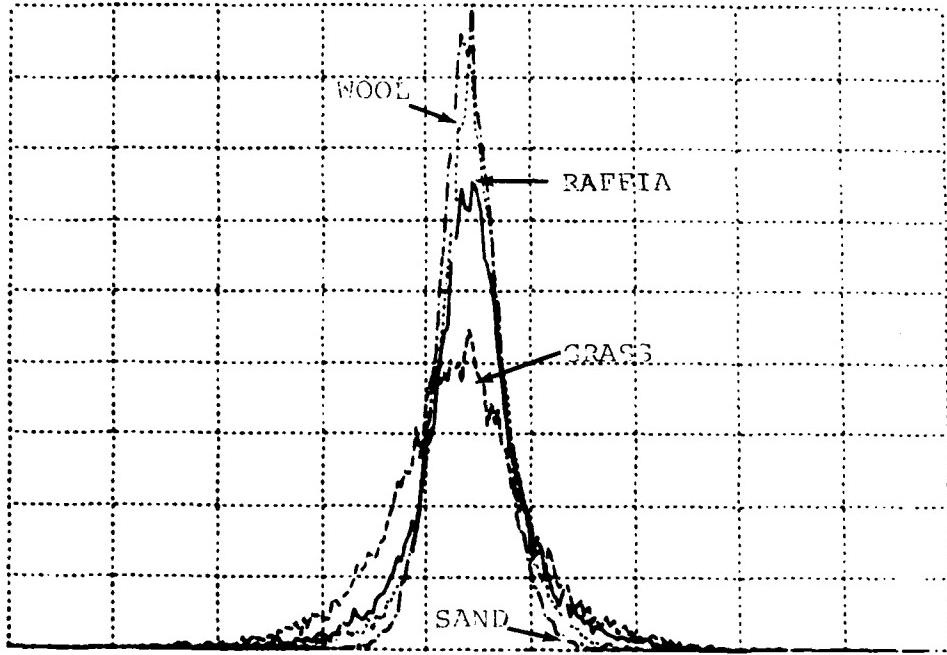


Figure 5. First order amplitude histograms of whitened natural texture fields

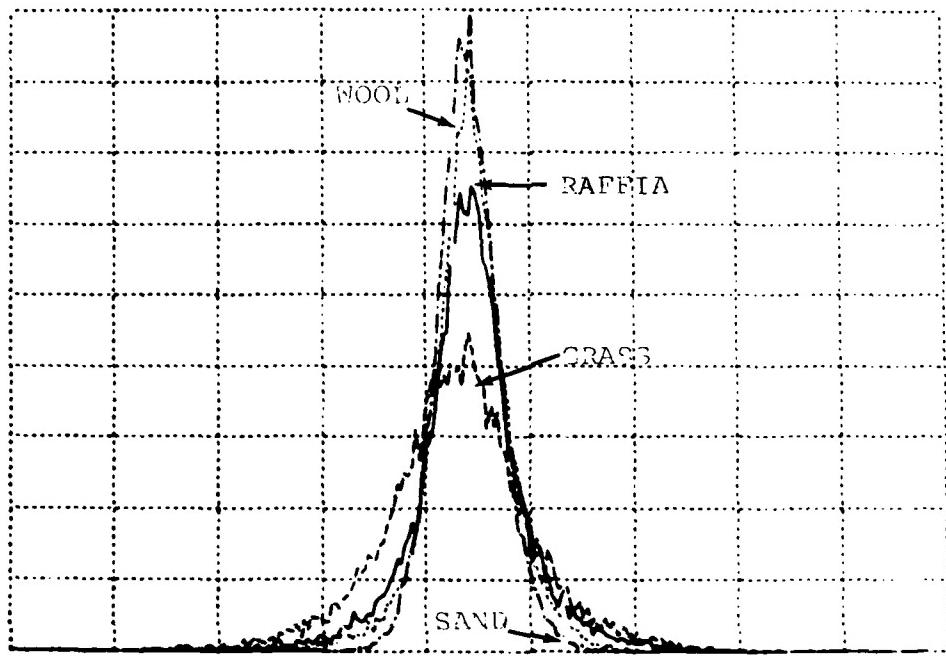


Figure 5. First order amplitude histograms of whitened natural texture fields

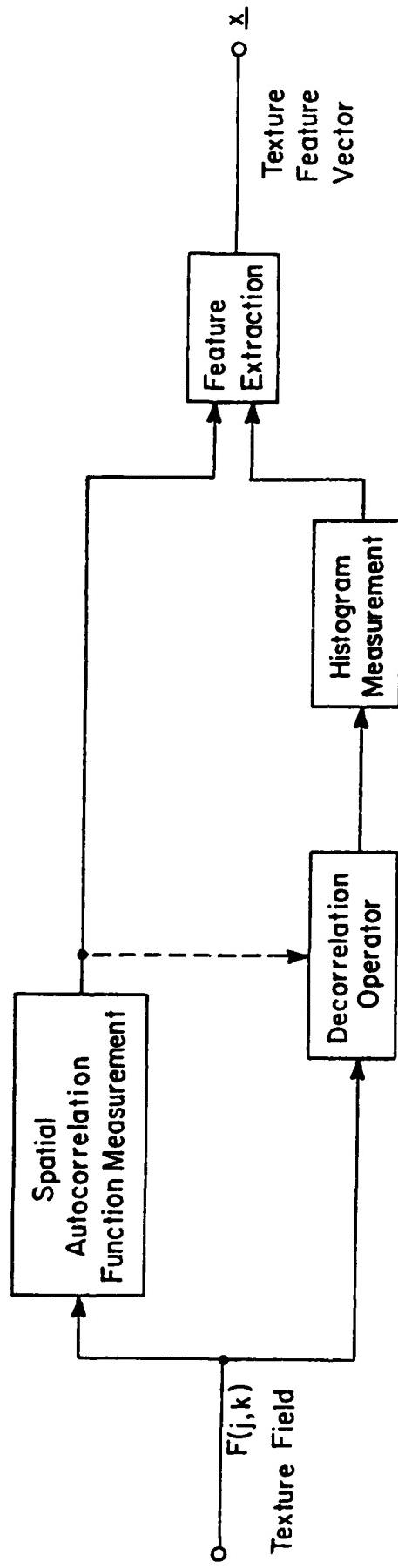


Figure 6. stochastic-based texture feature extraction method

spatial autocorrelation function and the decorrelated field histogram. The potential performance of these features for texture classification is considered in the following section.

A first order histogram $P(b)$ of L levels can be represented rather compactly by its central moments. The first four moments are defined below [15,p.472]:

average

$$b_A = \sum_{b=0}^{L-1} bP(b) \quad (5a)$$

deviation

$$b_D = \left[\sum_{b=0}^{L-1} (b-b_A)^2 P(b) \right]^{\frac{1}{2}} \quad (5b)$$

skewness

$$b_S = \frac{1}{b_D^3} \sum_{b=0}^{L-1} (b-b_A)^3 P(b) \quad (5c)$$

kurtosis

$$b_K = \frac{1}{b_D^4} \sum_{b=0}^{L-1} (b-b_A)^4 P(b) - 3 \quad (5d)$$

The factor 3 in the kurtosis expression is included so that

the kurtosis of a Gaussian histogram is zero.

The histogram moment called average tends toward zero for the whitening and Laplacian operators. For the Sobel gradient operator, it is a measure of the average gradient amplitude. The deviation parameter is near unity for a whitening transformation since this operator is designed to produce a unit variance decorrelated field. The skewness parameter measures the asymmetry of the histogram, and the kurtosis parameter provides an indication of the departure of the histogram from a Gaussian shape.

There are a number of methods that could be used to represent the spatial autocorrelation function. The method chosen, because of its simplicity, is representation by two-dimensional spread measures analogous to eq.(5). The general form of the autocorrelation function spread measure is

$$S(u, v) = \sum_{m=0}^T \sum_{n=0}^T (m - \eta_m)^u (n - \eta_n)^v \tilde{A}_F(m, n) \quad (6)$$

where

$$\eta_m = \sum_{m=0}^T \sum_{n=0}^T m \tilde{A}_F(m, n) \quad (7a)$$

$$\eta_n = \sum_{m=0}^T \sum_{n=0}^T n \tilde{A}_F(m, n) \quad (7b)$$

$$\tilde{A}_F(m, n) = \frac{A_F(m, n)}{\sum_{m=0}^T \sum_{n=0}^T A_F(m, n)} \quad (7c)$$

Features of potential interest include: the profile spreads, $S(2,0)$ and $S(0,2)$; the cross-relation, $S(1,1)$; and the second degree spread, $S(2,2)$.

Bhattacharyya Distance Figure of Merit

In this study, texture features have been evaluated according to their Bhattacharyya distance [16,p.268] figure of merit for texture prototypes. The Bhattacharyya distance (B-distance for simplicity) is a scalar function of the probability densities of features of two classes defined as

$$B(S_1, S_2) = -\ln \left\{ \int [p(\underline{x}|S_1)p(\underline{x}|S_2)]^{\frac{1}{2}} d\underline{x} \right\} \quad (8)$$

where \underline{x} denotes a feature vector with conditional density $p(\underline{x}|S_i)$ for class S_i . It can be shown [16,p.267] that the B-distance is monotonically related to the Chernoff bound of the probability of classification error using a Bayes classifier. The bound on the error probability is

$$P \leq [P(S_1)P(S_2)]^{\frac{1}{2}} \exp\{-B(S_1, S_2)\} \quad (9)$$

where $P(S_i)$ represents the a priori class probability. For future reference, the Chernoff error bound is tabulated in Table 1 as a function of B-distance for equally likely texture classes.

Table 1
Bhattacharyya Distance Versus Error Bound

$B(S_1, S_2)$	Error Bound
1	1.84×10^{-1}
2	6.77×10^{-2}
4	9.16×10^{-3}
6	1.24×10^{-3}
8	1.68×10^{-4}
10	2.27×10^{-5}
12	3.07×10^{-6}

For Gaussian densities, the B-distance becomes

$$B(S_1, S_2) = \frac{1}{8}(\underline{u}_1 - \underline{u}_2)^T \left[\frac{\Sigma_1 + \Sigma_2}{2} \right]^{-1} (\underline{u}_1 - \underline{u}_2) + \frac{1}{2} \ln \left\{ \frac{\left| \frac{1}{2}(\Sigma_1 + \Sigma_2) \right|}{|\Sigma_1|^{\frac{1}{2}} |\Sigma_2|^{\frac{1}{2}}} \right\} \quad (10)$$

where \underline{u}_i and Σ_i represent the feature mean vector and the feature covariance matrix of the classes, respectively. Calculation for other densities is generally difficult. Histogram measurements of the texture feature components

indicates that the Gaussian model is reasonable.

The B-distance has been computed for several feature vector sets of prototype texture fields. In these experiments, the natural stochastic texture fields of figure 1 have been subdivided into 64 non-overlapping prototype regions of 64x64 pixels. Texture features have been extracted from each region and formed into a texture feature vector. Next, the mean and covariance of the feature vector have been computed and substituted into eq.(10) to obtain the B-distance for pairs of prototype fields.

Table 2 contains a listing of B-distances for three texture feature sets that measure the shape of the autocorrelation function of each prototype field. With feature set 1, containing four features, the B-distances of the natural texture fields correspond to misclassification error bounds from about 6% to 20%. The B-distances are much smaller for feature sets 2 and 3 employing two features and one feature, respectively. The B-distance measurements of Table 2 indicate that autocorrelation shape features of texture fields, by themselves, are probably not adequate for texture classification.

Table 3 contains listings of B-distances for texture features consisting of histogram moments of whitened texture fields (set 1 to set 3) plus the texture field autocorrelation function shape (set 4 to set 7). For feature set 1, consisting of the first four histogram moments, the classification error bounds are less than 16% for the natural textures. The error bounds increase slightly for feature set 2 which uses only the skewness and kurtosis histogram moments. The error bounds using only kurtosis are quite high. These examples indicate that

Table 2
 Bhattacharyya Distance of Texture Feature Sets for Prototype
 Texture Fields Autocorrelation Features

FIELD	PAIRS	SET #1	SET #2	SET #3
		(4)	(2)	(1)
GRASS	SAND	1.15	0.72	0.70
GRASS	RAFFIA	2.10	1.65	1.18
GRASS	WOOL	0.97	0.52	0.01
SAND	RAFFIA	0.92	0.65	0.21
SAND	WOOL	1.72	0.67	0.61
RAFFIA	WOOL	2.78	1.70	1.09
AVERAGE		1.61	0.98	0.63

SET #1: S(2,0), S(0,2), S(1,1), S(2,2)

SET #2: S(1,1), S(2,2)

SET #3: S(2,2)

Table 3
 Bhattacharyya Distance of Texture Feature Sets
 for Prototype Texture Fields-Whitened Field
 Histogram and Autocorrelation Features

FIELD PAIRS		TEXTURE FEATURE SETS						
		Set #1	Set #2	Set #3	Set #4	Set #5	Set #6	Set #7
Grass	Sand	4.394	4.285	0.713	5.647	5.528	5.075	5.025
Grass	Raffia	1.154	1.042	0.525	3.332	3.221	2.710	2.255
Grass	Wool	1.682	1.595	0.144	2.773	2.614	2.195	1.632
Sand	Raffia	12.089	11.936	0.264	13.698	13.523	12.965	12.358
Sand	Wool	11.758	11.617	1.911	13.391	13.264	12.341	12.229
Raffia	Wool	4.027	3.890	1.476	7.302	7.140	5.769	5.133
AVERAGE (EXCLUDING 1A-1B)		5.85	5.73	0.84	7.69	7.55	6.84	6.44

SET #1: b_A, b_D, b_S, b_K

SET #2: b_S, b_K

SET #3: b_K

SET #4: $b_A, b_D, b_S, b_K, S(2,0), S(0,2), S(1,1), S(2,2)$

SET #5: $b_S, b_K, S(2,0), S(0,2), S(1,1), S(2,2)$

SET #6: $b_S, b_K, S(1,1), S(2,2)$

SET #7: $b_S, b_K, S(2,2)$

texture features based upon the histogram shape of a whitened texture field, by themselves, may be adequate for natural texture fields.

Feature sets 4 to 7 of Table 2 combine histogram and autocorrelation shape measures. Set 4 containing four histogram and four autocorrelation features provides quite high B-distances for the natural textures. The B-distances are still quite high for the three element feature set 3 containing shape measures b_S , b_K and $S(2,2)$. Therefore, on the basis of these experiments, the texture feature extraction method of figure 6 combining autocorrelation function measurement of a texture field coupled with histogram measurement of the whitened texture field, offers a viable means of texture classification.

Summary

Bounds have been established for necessary and sufficient parameters for characterization of stochastic texture fields. This information, coupled with a stochastic model of texture field generation, has led to the development of a texture feature extraction method based on representation of the autocorrelation function of a texture field plus the gray scale histogram of a decorrelated version of the texture field. Feature representation is in terms of central moments of the autocorrelation function and the histogram. The feature vector so obtained has been evaluated by Bhattacharyya distance measurements. Testing with prototype texture fields indicates that large Bhattacharyya distances can be obtained between texture field pairs with the stochastic-based feature extraction method.

Acknowledgement

Mr. Andre Gaglowicz of the Institut de Recherche d'Informatique et d'Automatique, Le Chesney, France contributed greatly to the research study [9] leading to this paper. His contribution is gratefully acknowledged.

References

1. R.M. Pickett, "Visual Analysis of Texture in the Detection and Recognition of Objects," in Picture Processing and Psychopictorics, B.S. Lipkin and A. Rosenfeld, Eds., Academic Press, New York, pp. 289-308, 1970.
2. J.K. Hawkins, "Textural Properties for Pattern Recognition," in Picture Processing and Psychopictories, B.S. Lipkin and A. Rosenfeld, Eds., Academic Press, New York, pp. 347-370, 1970.
3. B. Julesz, "Visual Pattern Discrimination," IRE Transactions Information Theory, Vol. IT-8, no. 1, pp. 84-92, February 1962.
4. B. Julesz, Foundations of Cyclopean Perception, University of Chicago Press, 1970.
5. B. Julesz et al., "Inability of Humans to Discriminate Between Visual Textures that Agree in Second-Order Statistics-Revisited," Perception, Vol. 2, pp.391-405, 1973.
6. b. Julesz, "Experiments in the Visual Perception of Texture," Scientific American, Vol. 232, No.4, pp.34-43, April 1975.
7. I. Pollack, Perceptual Psychophysics, Vol. 13, pp.276-280, 1973.

8. S.R. Purks and W. Richards, "Visual Texture Discrimination Using Random Dot Patterns," Journal Optical Society of America, Vol. 67, no.6, pp.765-771, June 1977.
9. W.K. Pratt, O.D. Faugeras, and A. Gagalowicz, "Visual Discrimination of Stochastic Texture Fields," IEEE Transactions on Systems, Man, and Cybernetics, November 1978.
10. P. Brodatz, Texture: A Photograph Album for Artists and Designers, Dover, New York, 1956.
11. R.M. Haralick, K. Shanmugan, and I. Dinstein, "Textural Features for Image Classification," IEEE Transactions Systems, Man, and Cybernetic, Vol. SMC-3, No.6, pp.610-621, November 1973.
12. R.M. Haralick and K. Shanmugan, "Computer Classification of Reservoir Sandstones," IEEE Transactions Geoscience Electronics, Vol. GE-11, pp.171-177, October 1973.
13. J.S. Weska, C.R. Dyer, and A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," IEEE Transactions Systems, Man, and Cybernetics, Vol. SMC-6, No.4, pp.269-285, April 1976.
14. R.W. Conners, "Some Theory on Statistical Models for Texture and Its Application to Radiographic Image Processing," Ph.D. Thesis, College of Engineering, University of Missouri-Columbia, December 1976.
15. W.K. Pratt, Digital Image Processing, Wiley-Interscience, New York, 1978.

16. K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972.

2.2 Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors

Ikram E. Abdou and William K. Pratt

Introduction

Quantitative design and performance evaluation techniques have been developed for the enhancement/thresholding class of image edge detectors. The design techniques are based on statistical detection theory and deterministic pattern recognition classification procedures. The performance evaluation methods developed include: (a) deterministic measurement of the edge gradient amplitude; (b) comparison of the probabilities of correct and false edge detection; and (c) figure of merit computation.

Enhancement/Thresholding Luminance Edge Detectors

The edge enhancement/thresholding edge detection method is described in figure 1 [1]. In this method, the discrete image array $F(j,k)$ is spatially processed by a set of N linear operators or masks $H_i(j,k)$ to produce a set of gradient functions

$$G_i(j,k) = F(j,k) \otimes H_i(j,k) \quad (1)$$

where \otimes denotes two-dimensional spatial convolution. Next, at each pixel, the gradient functions are combined by a

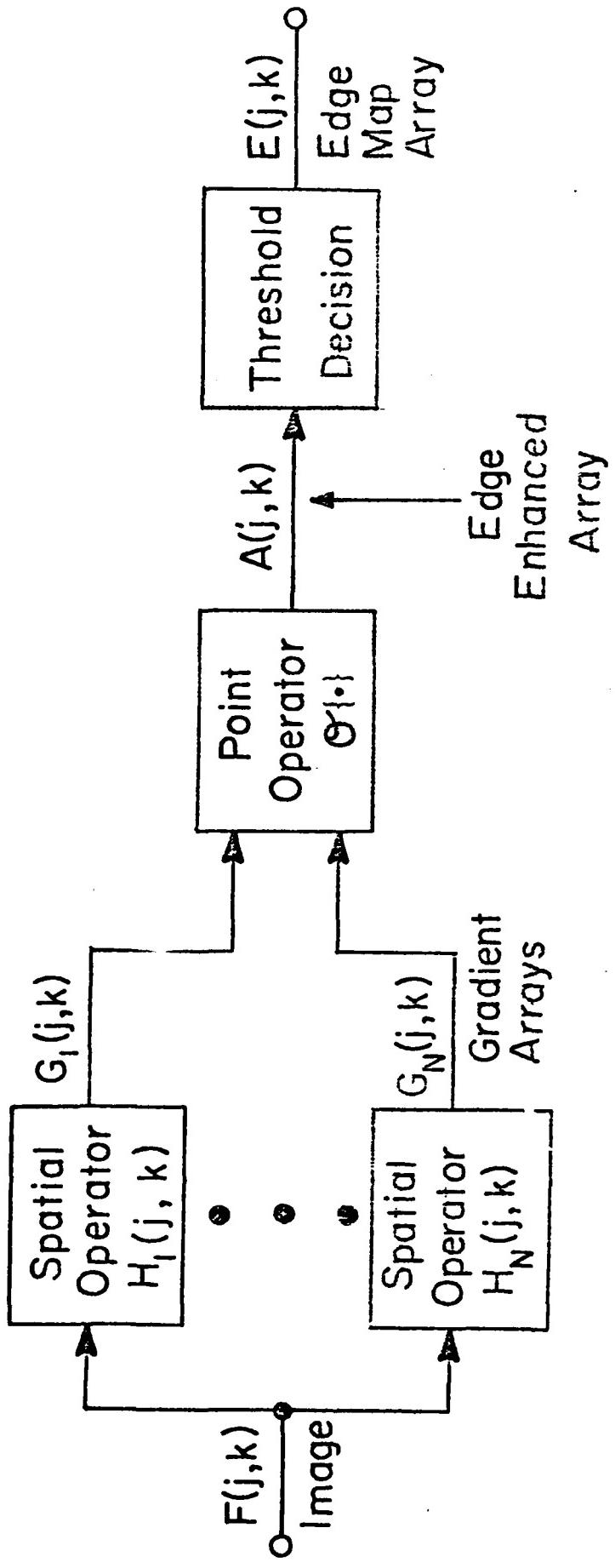


Figure 1. Edge enhancement/thresholding edge detection system

linear or nonlinear point operator $\mathcal{O}\{\cdot\}$ to create an edge enhanced array

$$A(j,k) = \mathcal{O}\{G_i(j,k)\} \quad (2)$$

Typical forms of the point operator include the root mean square, magnitude, and maximum. The enhanced array $A(j,k)$ provides a measure of the edge discontinuity at the center of the gradient mask. An edge decision is formed on the basis of the amplitude of $A(j,k)$ with respect to a threshold (t). If $A(j,k) \geq t$, an edge is assumed present, and if $A(j,k) < t$, no edge is indicated. The edge decision is usually recorded as a binary edge map $E(j,k)$ where a one value indicates an edge and a zero value, no edge. There are two types of spatial edge enhancement operators: the differential and the template matching operators.

Differential Operators. The differential operators perform discrete differentiation of an image array to produce a gradient field. This group includes the Roberts [2], Prewitt [3], and Sobel [4,p.271] operators. The Roberts operator is a 2×2 pixel mask in which

$$H_1 = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (3a)$$

$$H_2 = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad (3b)$$

The Prewitt and Sobel operators are 3×3 pixel operators where

$$H_1 = \begin{bmatrix} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{bmatrix} \quad (4a)$$

$$H_2 = \begin{bmatrix} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{bmatrix} \quad (4b)$$

With the Prewitt operator, $c = 1$ and with the Sobel operator, $c = 2$. These operators usually utilize a root mean square point nonlinearity to produce an edge enhanced array

$$A(j,k) = \left([G_1(j,k)]^2 + [G_2(j,k)]^2 \right)^{\frac{1}{2}} \quad (5a)$$

A magnitude point nonlinearity yielding

$$A(j,k) = |G_1(j,k)| + |G_2(j,k)| \quad (5b)$$

is often used for computational simplicity.

Edge orientation can be obtained from the relationship between the horizontal and vertical gradient functions. For the 2×2 operators, the edge orientation angle $\theta(j,k)$, with respect to the horizontal axis, is defined to be

$$\theta(j,k) = \frac{\pi}{4} + \tan^{-1} \left[\frac{G_2(j,k)}{G_1(j,k)} \right] \quad (6a)$$

and for 3×3 operators

$$\theta(j,k) = \tan^{-1} \left[\frac{G_2(j,k)}{G_1(j,k)} \right] \quad (6b)$$

Template Matching Operators. The template matching operators are a set of masks representing discrete approximations to ideal edges of various orientation.

Figure 2 gives several examples for two of eight possible compass orientations. These operators include the compass gradient introduced by Prewitt [3], the Kirsch [5], and the 3-level and 5-level template masks. The latter two operators are related to the Prewitt and Sobel differential operators, respectively. With these operators, the enhancement is formed as the maximum of the gradient arrays. Thus

$$A(j,k) = \max_i \{ |G_i(j,k)| \} \quad (7)$$

The edge orientation $\theta(j,k)$ corresponds to the compass direction of the largest gradient.

Edge Detector Sensitivity Analysis

Simple geometric calculations can be performed to determine the edge gradient and detected edge orientation response as a function of actual edge orientation. Results of these calculations are presented in figures 3 and 4. The curves indicate that the Prewitt and Sobel square root differential operators and the template matching operators all possess an amplitude response relatively invariant to actual edge orientation. The Sobel operator provides the most linear response between actual and detected edge orientation.

Statistical Design Procedure

Edge detection can be regarded as a hypothesis testing problem to determine if an image region contains an edge or contains no edge. Let $P(\text{edge})$ and $P(\text{no-edge})$ denote the a priori probabilities of these events. Then, the edge detection process can be characterized by the probability of correct edge detection

1	1	-1
1	-2	-1
1	1	-1

i) Compass
gradient

3	3	-5
3	0	-5
3	3	-5

ii) Kirsch

1	0	-1
1	0	-1
1	0	-1

iii) 3-level

1	0	-1
2	0	-2
1	0	-1

iv) 5-level

(b) mask H_1

1	-1	-1
1	-2	-1
1	1	1

i) Compass
gradient

3	-5	-5
3	0	-5
3	3	3

ii) Kirsch

0	-1	-1
1	0	-1
1	1	0

iii) 3-level

0	-1	-2
1	0	-1
2	1	0

iv) 5-level

(c) mask H_2

Figure 2. Template matching operators

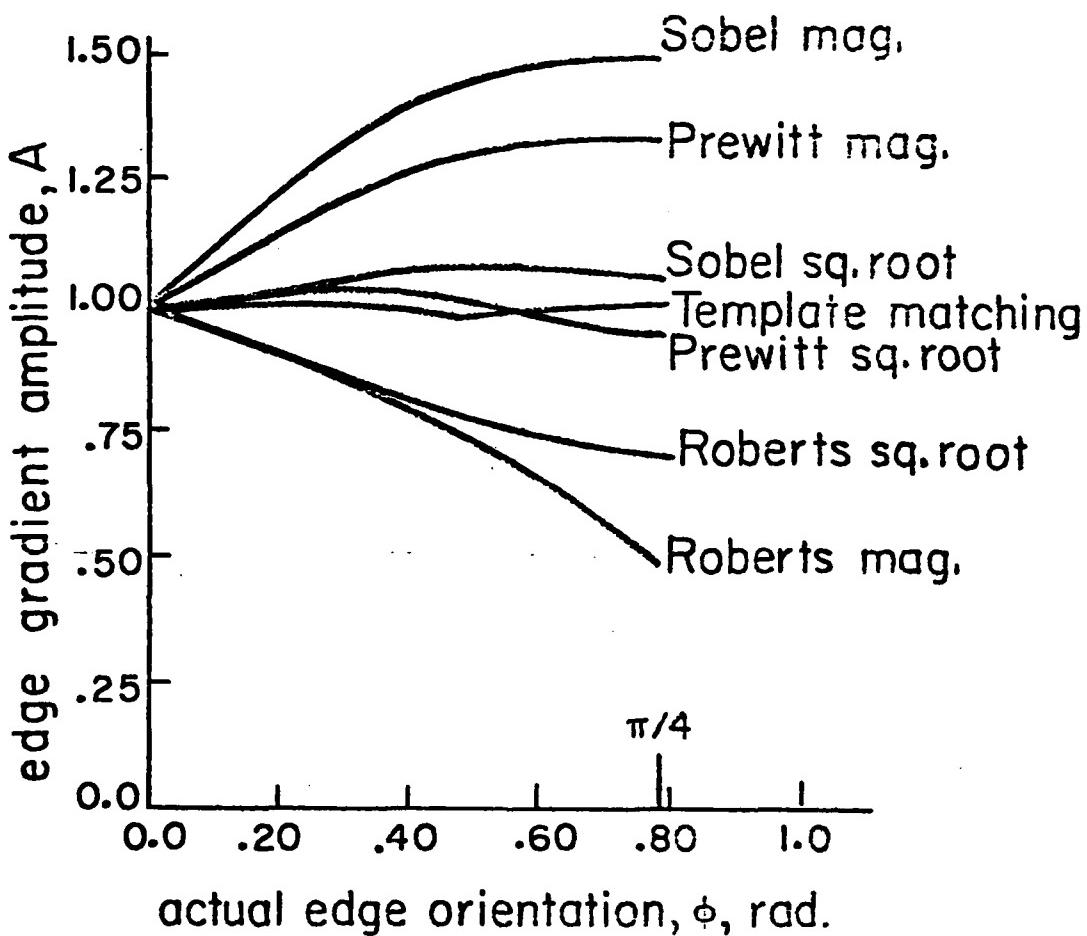


Figure 3. Edge gradient amplitude response as a function of actual edge orientation for 2×2 and 3×3 operators

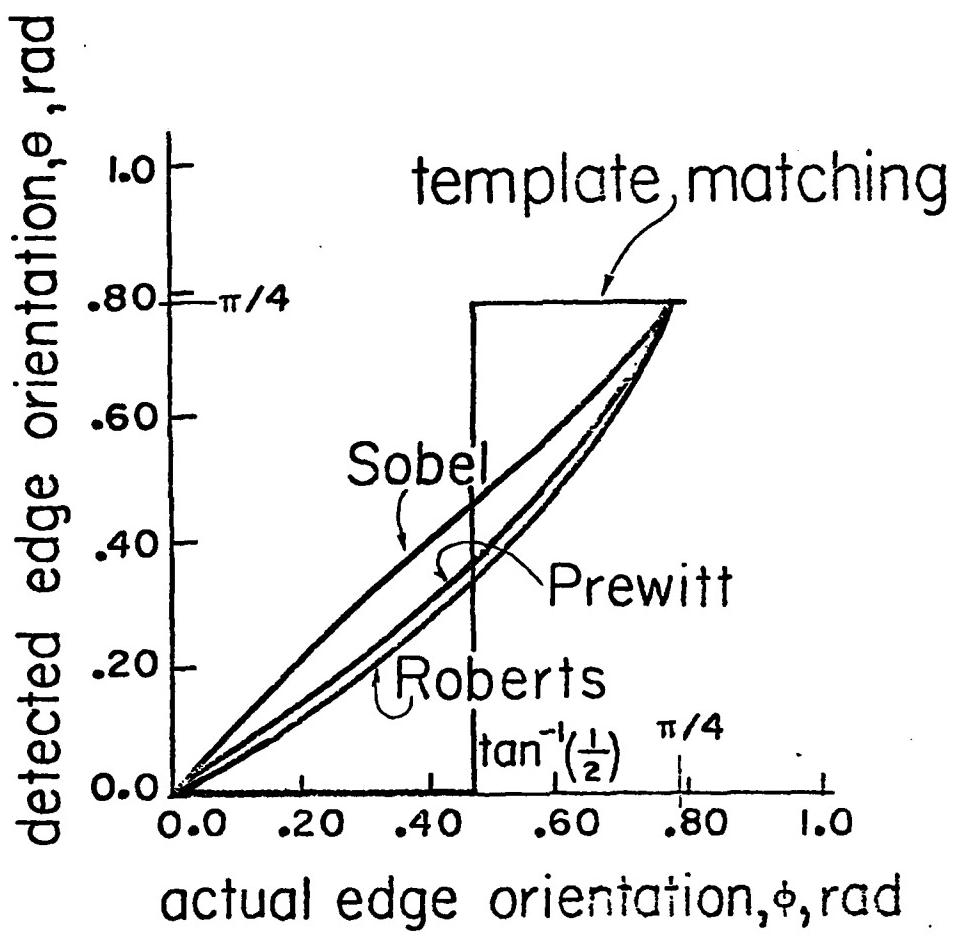


Figure 4. Detected edge orientation as a function of actual size orientation for 2×2 and 3×3 operators

$$P_D = P(A \geq t | \text{edge}) = \int_t^{\infty} p(A | \text{edge}) dA \quad (8)$$

and the probability of false edge detection

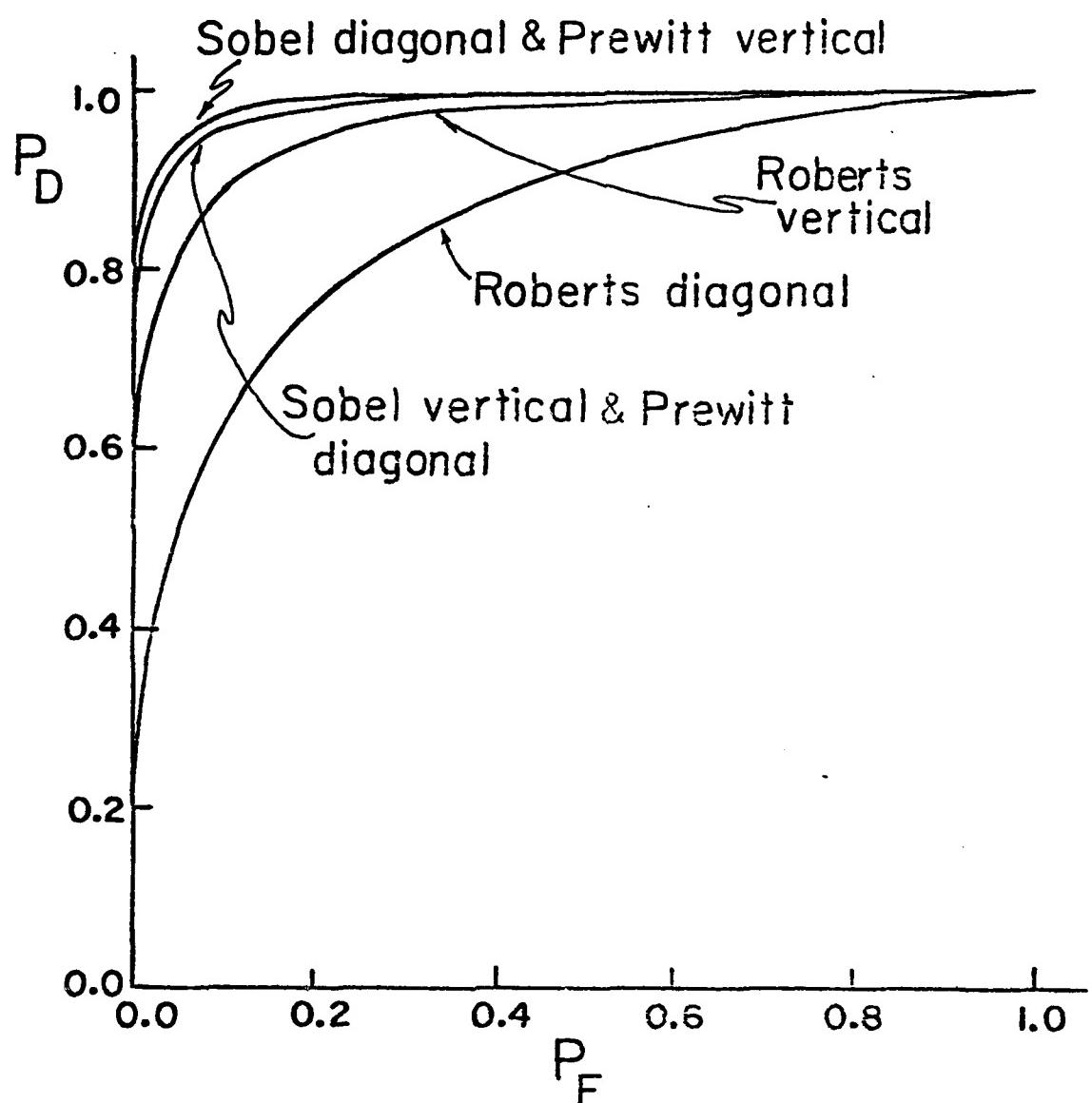
$$P_F = P(A \geq t | \text{no-edge}) = \int_t^{\infty} p(A | \text{no-edge}) dA \quad (9)$$

where (t) is the edge decision threshold and $p(A | \text{edge})$ and $p(A | \text{no-edge})$ are the conditional probability densities of the edge enhanced field $A(j,k)$.

The detection performance of edge detectors can be readily compared by a parametric plot of the correct detection probability P_D versus false detection probability P_F in terms of the detection threshold (t) . Figure 5 presents such plots for square root differential operators and template matching operators for vertical and diagonal edges and a signal-to-noise ratio (SNR) of 10.0. From these curves, it is apparent that the Sobel and Prewitt 3×3 operators are superior to the Roberts 2×2 operators. The Prewitt operator is better than the Sobel operator for a vertical edge. But, for a diagonal edge, the Sobel operator is superior. In the case of template matching operators, the 3-level and 5-level operators exhibit almost identical performance that is superior to the Kirsch and compass gradient operators. Finally, the Sobel and Prewitt differential operators perform slightly better than the 3-level and 5-level template matching operators.

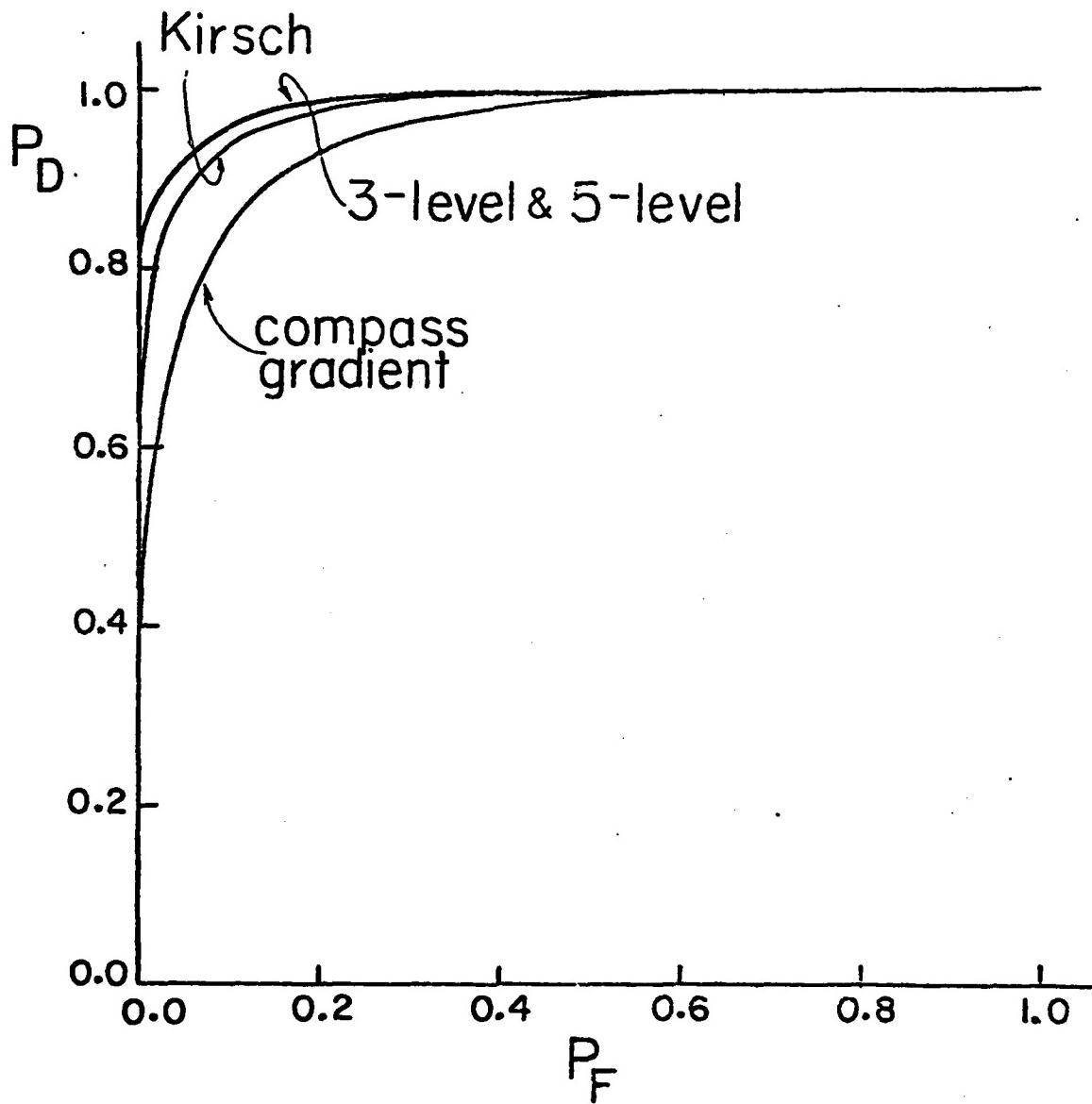
Pattern Classification Design Procedure

There are two difficulties with the statistical design procedure described in the previous section: reliability of



(a) differential operators, SNR=10

Figure 5. Probability of detection versus probability of false detection for 2×2 and 3×3 operators



(b) template operators, SNR=10

Figure 5. (continued)

the stochastic edge model and analytic problems associated with complex edge models such as non-Gaussian signal-dependent noise. The pattern classification design procedure described in this section avoids these difficulties.

Edge detection can be viewed as a classical pattern recognition or classification problem. A pattern consisting of the pixels encompassed by an edge detection operator is classified as a region containing an edge or no-edge on the basis of an extracted region feature, the amplitude $A(j,k)$ of the edge enhancement. Classification can be accomplished by the linear discriminant function method [8] in which the edge hypothesis is selected if

$$\underline{w}^T \underline{x} > 0 \quad (10a)$$

and rejected if

$$\underline{w}^T \underline{x} < 0 \quad (10b)$$

$\underline{w} = [w(1), w(2)]$ where $w(1)$ and $w(2)$ are weighting factors of the weight vector and $\underline{x} = [A, 1]$. The weight factors are related to the decision threshold by

$$t = -\frac{w(2)}{w(1)} \quad (11)$$

Components of \underline{w} can be determined by the Ho-Kashyap training procedure [6,9] using a set of prototype pixel regions containing edges or no-edges.

An experiment has been performed to evaluate the pattern classification edge detector design procedure. In this experiment, sets of 20 edge prototypes and 20 no-edge

prototypes have been generated for vertical and diagonal edges embedded in independent Gaussian noise at signal-to-noise ratios of 1.0 and 10.0. This prototype data has then been used to determine the optimum threshold. After the training phase was completed, the edge detectors were tested with 250 other prototypes. Optimum thresholds and detection probabilities are tabulated in Table 1 for various edge detectors. It is interesting to note, that in most cases, the optimum threshold converged to a value for which the error probabilities were approximately equal ($P_F \approx 1 - P_D$). This is the same result that is obtained by the Bayes minimum error design procedure if edges and no edges are equally probable. Thus, in the Gaussian noise case, similar design results are obtained with the statistical and pattern classification design approaches.

Figure of Merit Comparison

The probabilities of correct detection and false detection, obtained analytically or experimentally, are useful performance indicators for edge detectors. However, these detection probability functions do not distinguish between the various types of errors that can be introduced by an edge detector. Pratt [1,p.495] has developed a simple figure of merit for edge detectors that provides a relative penalty for fragmented, smeared, and offset edges.

Pratt's figure of merit measurement procedure utilizes a square array of pixels with a vertically oriented ramp edge in its center. The edge parameters and noise level can be varied to generate test edges which are then processed by an edge detector to produce binary edge maps. The figure of merit is defined as

TABLE 1

Threshold Level and Error Probabilities for Ho-Kashyap Design Procedure

OPERATOR	Vertical Edge, SNR=1				Vertical Edge, SNR=10				Diagonal Edge, SNR=1				Diagonal Edge, SNR=10							
	experiment		theory		experiment		theory		experiment		theory		experiment		theory					
	\tilde{t}	P_D	P_F	P_D	P_F	\tilde{t}	P_D	P_F	\tilde{t}	P_D	P_F	\tilde{t}	P_D	P_F	\tilde{t}	P_D	P_F			
Roberts	1.36	52.0	37.6	55.89	39.87	0.67	91.2	11.6	89.16	10.47	1.74	55.60	46.80	55.10	46.88	0.78	74.80	20.80	77.79	22.11
Square Root																				
Roberts Magnitude	1.22	52.0	38.4	55.16	39.30	0.62	90.8	7.6	89.20	9.85	2.24	54.40	46.40	53.85	45.70	0.97	75.60	19.60	76.81	23.30
Sobel	1.18	60.8	41.2	60.01	39.54	0.66	92.0	9.6	92.34	5.69	1.14	63.20	39.20	60.40	37.63	0.63	90.80	9.20	94.65	5.28
Square Root																				
Prewitt	1.16	59.5	36.6	60.80	38.40	0.66	93.0	3.8	91.20	4.80	1.19	61.20	39.60	59.27	38.71	0.64	90.00	8.40	93.07	6.42
Square Root																				
Compass	1.52	57.6	45.2	61.27	46.56	0.73	85.2	12.8	88.58	13.55	1.51	57.60	46.80	61.80	47.20	0.71	80.80	14.00	90.00	15.30
Gradient																				
Kirsch	1.43	56.0	38.4	53.08	34.08	0.69	89.2	9.2	89.78	5.76	1.45	54.40	36.00	52.40	32.40	0.79	82.80	3.60	82.50	2.30
3-Level	1.16	60.8	38.4	59.02	36.92	0.65	89.6	6.4	92.64	3.79	1.16	59.20	38.40	58.70	36.50	0.61	89.60	8.40	94.60	5.60
5-Level	1.24	58.0	37.6	58.12	36.09	0.66	90.8	6.8	92.45	4.90	1.22	60.40	39.20	59.30	37.40	0.65	90.00	8.40	93.10	5.40

$$F = \frac{1}{\max\{I_I, I_A\}} \sum_{i=1}^{I_A} \frac{1}{1+\alpha d^2(i)} \quad (12)$$

where I_I and I_A are the number of ideal and actual edge points, $d(i)$ is the pixel miss distance of the i -th edge detected, and α is a scaling constant chosen to be $\alpha = 1/9$ to provide a relative penalty between smeared edges and isolated, but offset, edges. This technique can be extended to diagonal edges.

Figure 6 contains a figure of merit comparison between the best differential and template matching operators. Photographs of detector edge maps corresponding to several of the data points are presented in figure 7. The curves indicate that among the class of differential operators, the Prewitt and Sobel operators provide a substantially higher figure of merit than the Roberts operator. The Prewitt operator exhibits a somewhat larger figure of merit than the Sobel operator for a vertical edge, while for a diagonal edge, their performances are nearly the same. For the template operators, the 3-level, 5-level, and Kirsch operators are clearly superior to the compass gradient operator. The 3-level operator is dominant by a slight margin at all signal-to-noise ratios for diagonal edges, but for vertical edges the relative dominance changes with signal-to-noise ratio. The Prewitt square root differential operator gives a slightly higher figure of merit than the 3-level template matching operator for vertical edges. For diagonal edges, the reverse is true.

Summary

On the basis of quantitative design and evaluation techniques presented, the following conclusions have been

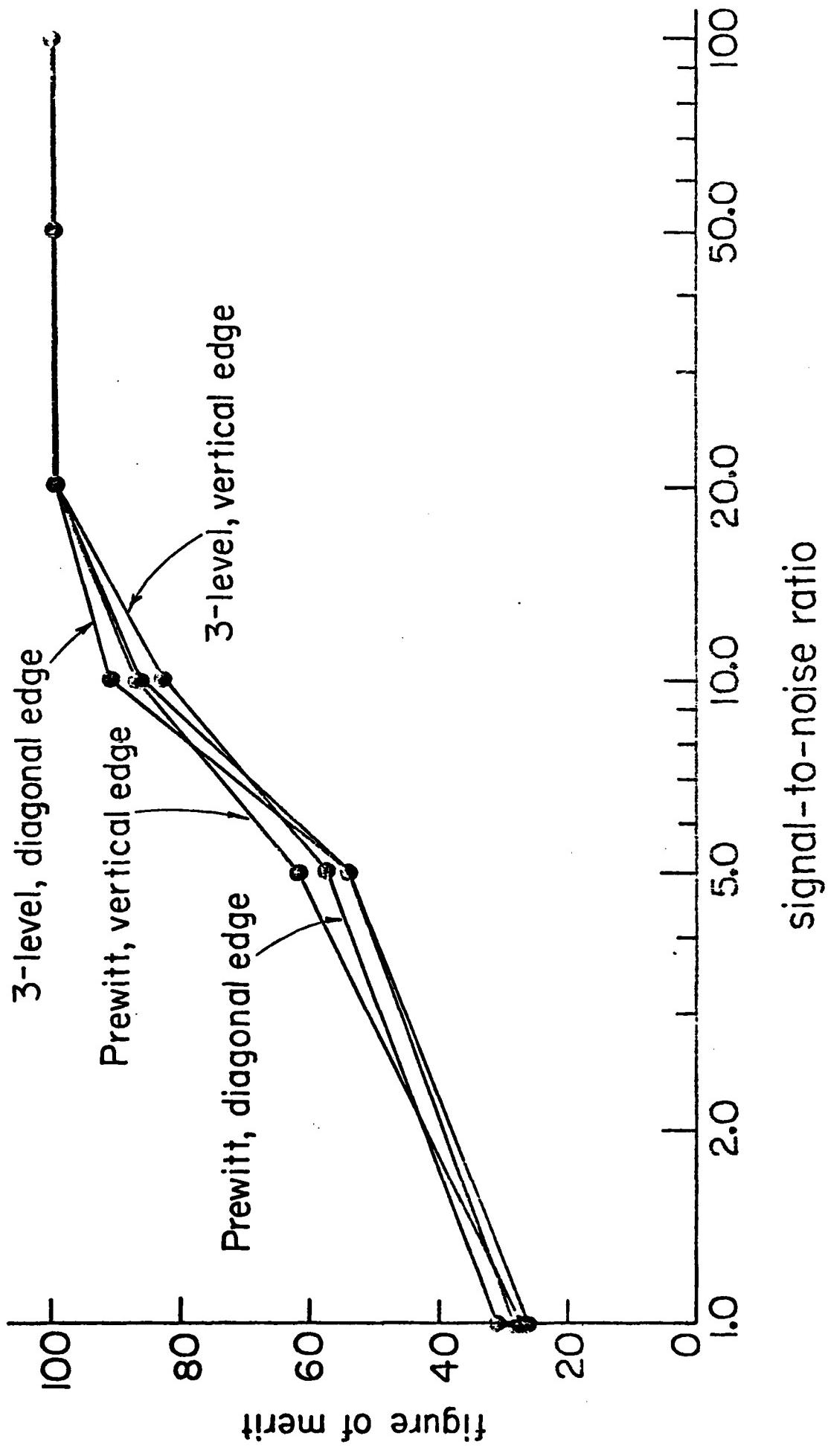


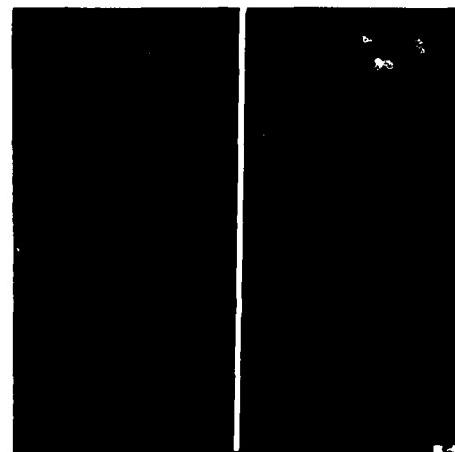
Figure 6. Figure of merit comparison between differential and template matching operators



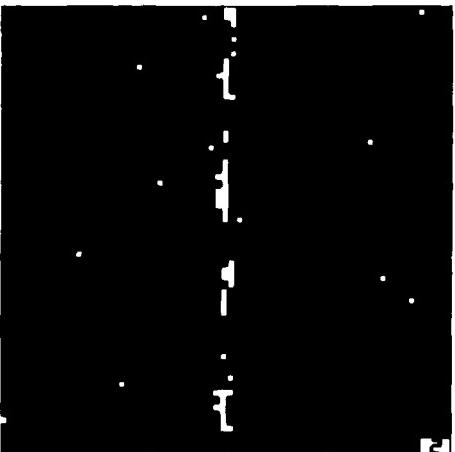
(a) Prewitt square root
vertical edge, SNR=1



(b) Prewitt square root
vertical edge, SNR=10



(c) Prewitt square root
vertical edge, SNR=100



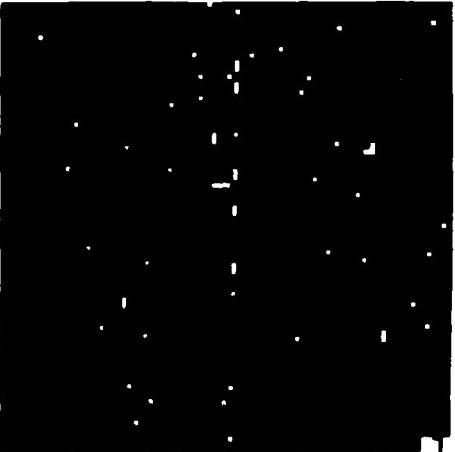
(d) Prewitt magnitude
vertical edge, SNR=10



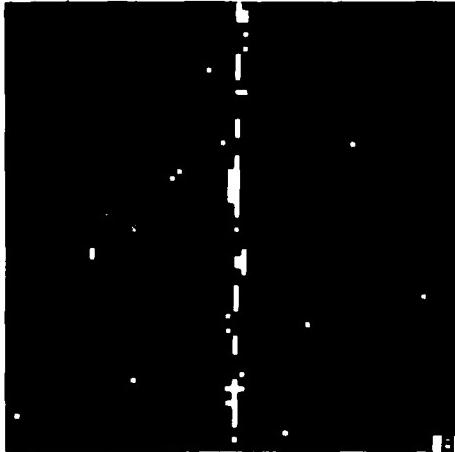
(e) Prewitt square root
diagonal edge, SNR=10



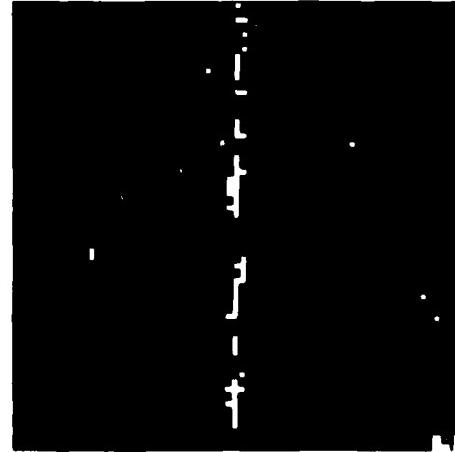
(f) Sobel square root
vertical edge, SNR=10



(g) Roberts square root
vertical edge, SNR=10



(h) 3-level
vertical edge, SNR=10



(i) Kirsch
vertical edge, SNR=10

Figure 7. Edge maps for 2×2 and 3×3 operators.

formulated.

1. The 3×3 differential edge detectors perform appreciably better than the 2×2 differential edge detectors.
2. The 3×3 Prewitt and Sobel differential edge detectors are the best of the 3×3 pixel differential class of edge detectors.
3. The 3-level edge detector is the best of the 3×3 pixel template matching class of edge detectors.
4. The 3×3 pixel 3-level template matching edge detector and the 3×3 pixel Sobel and Prewitt differential edge detectors perform almost equally well as a function of edge orientation and signal-to-noise ratio. It should be noted that differential edge detectors require fewer operations than template edge detectors.

References

1. W.K. Pratt, Digital Image Processing, Wiley-Interscience, New York, 1978.
2. L.G. Roberts, "Machine Perception of Three-Dimensional Solids," in Optical and Electro-Optical Information Processing, J.T. Tippett et al., eds., Massachusetts Institute of Technology Press, Cambridge, Massachusetts, 1965, pp. 159-197.
3. J.M.S. Prewitt, "Object Enhancement and Extraction," in Picture Processing and Psychopictorics, B.S. Lipkin and A. Rosenfeld, eds., Academic Press, New York, 1970.
4. R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.

5. R. Kirsch, "Computer Determination of the Constituent Structure of Biological Images," Computers and Biomedical Research, Vol. 4, No. 3, 1971, pp. 315-328.
6. I. Abdou, "Quantitative Methods of Edge Detection," University of Southern California, Image Processing Institute, USCIPI Report 800, Los Angeles, California, 1978.
7. K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972.
8. H.C. Andrews, Introduction to Mathematical Techniques in Pattern Recognition, Wiley-Interscience, New York, 1972.
9. Y.C. Ho and R.L. Kashyap, "An Algorithm for Linear Inequalities and Its Applications," IEEE Transactions Electronics Computers, Vol. EC-14, No. 5, October 1965, pp. 683-688.

2.3 Optical Psuedocolor Encoding of One-Dimensional Texture Patterns

Timothy C. Strand and David D. Garber

Introduction

Another paper [2] in this report illustrates the method by which various one-dimensional binary textures may be generated. These texture patterns may be color encoded according to their local spatial frequency contents. The results indicate that an optical spatial filtering system can be used to perform a simple type of texture-to-color conversion. Such a system could be used to enhance textural

differences for a human observer in general image scenes.

Basic Concepts

The application of pseudocolor to an image involves the introduction of color into a monochrome image where the color is used to encode additional information related to the image [1]. In the past it has been used almost exclusively to encode gray-level information. The advantage of this particular application is that the number of colors a human observer can differentiate is much larger than the number of intensity levels he can distinguish. Thus the addition of color can be used to effectively increase the amount of gray-level information available to the observer. Color encoding of local spatial frequency content is done using an optical spatial-filtering setup with a color filter placed in the spatial frequency plane. The system can be thought of as a multi-channel filtering system, where the separate channels are color coded. The advantages of the use of color to encode the spatial frequency information is that the output is in a form easily interpreted by either a human observer or a machine. Previous research [1], [3] has shown that spatial frequency pseudocolor could be a useful technique in digital image processing. However, this process is quite cumbersome to carry out on a computer since it typically requires one forward Fourier transform and three inverse Fourier transforms, one for each color. Thus, an optical processing system encoding spatial frequency content in real time provides many advantages over a digital approach.

The Experiment

The experimental setup is basically that of Fig. 1. For the texture experiments described here the illumination

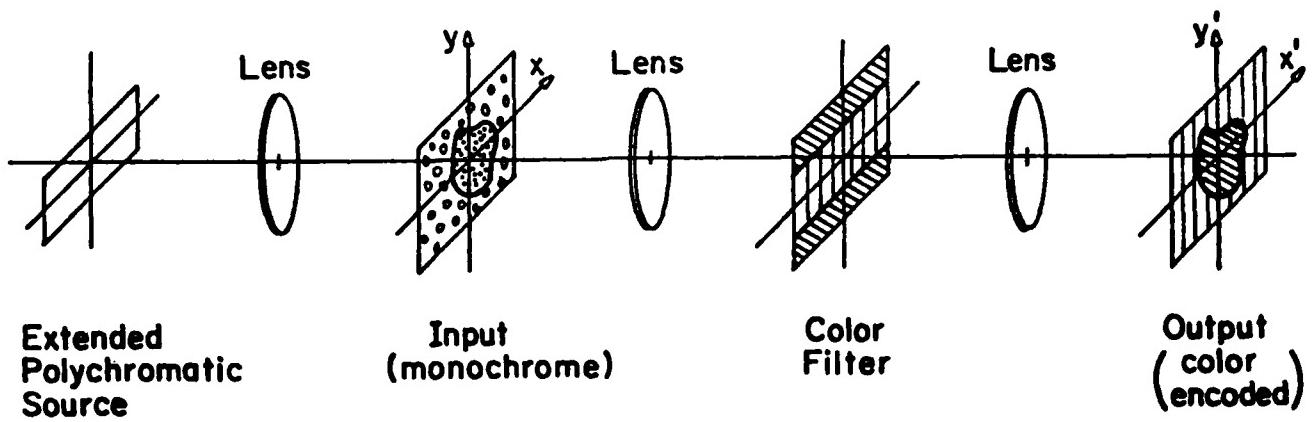


Figure 1. Optical spatial frequency pseudocolor system. A color filter in the pupil plane color encodes information on the spatial frequency distribution in the input.

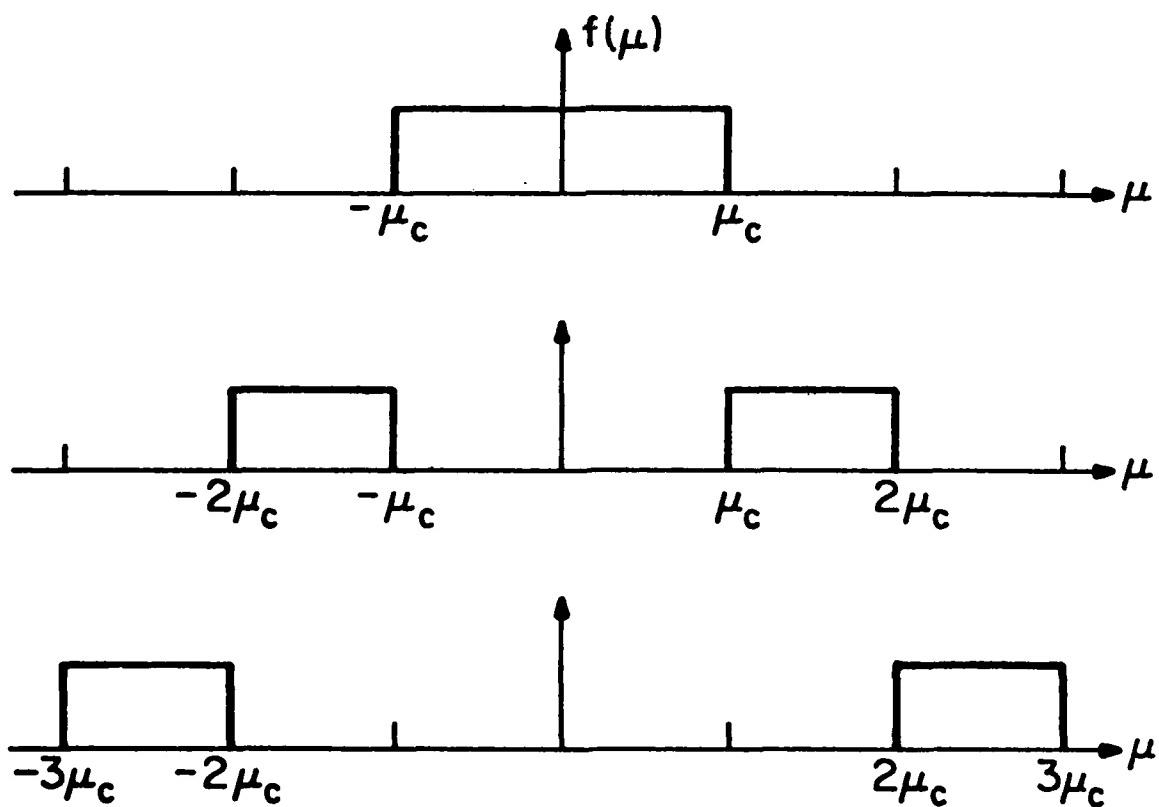


Figure 2. One-dimensional, three-color filter in the pupil plane. The coordinate μ represents spatial frequency.

used was derived from a tungsten filament lamp. Figure 3 shows the filter used for one-dimensional processing. It consists of the blue Wratten 47B, the green Wratten 58, and the red Wratten 92. This combination was selected after limited experimentation because it gave a good visual color balance. The optimal choice of colors depends upon the illumination, the detector, the application, and the texture spatial frequencies.

In order to confirm the theoretical concept of the system, (additional details concerning the system are given by J. Bescos and T.C. Strand [4]) the one-dimensional filter of Fig. 3 was made to correspond to the theoretical filter in Fig. 2. Color 1, the low frequency filter, was blue. The intermediate frequency region corresponding to color 2 was green. The high frequency region of color 3 was red.

Results

Texture is a complex image attribute that is difficult even to define precisely. Texture has been the subject of much research both in the fields of image processing and in visual perception. Although no simple method has been found to characterize all the facets of texture, it is generally recognized that there is a close connection between the texture and the spatial frequency content of a given image segment. This connection has been most recently studied by Purks and Richards [5]. They generated a series of binary texture patterns. They varied the different parameters controlling their artificial textures and measured the visual discriminability of these various textures. From this work Purks and Richards conclude, among other things, that differences in the spatial frequency content of their binary patterns correlate closely to the relative visual

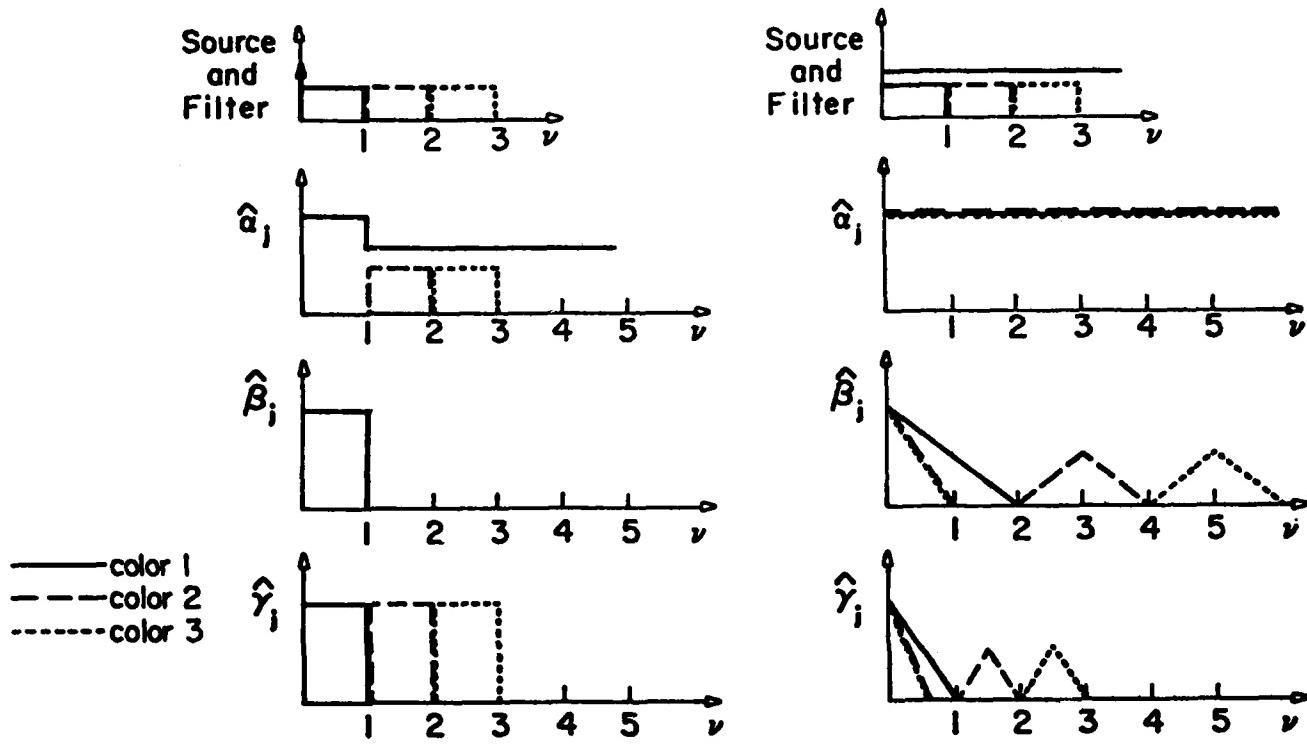


Figure 3. One-dimensional color filter corresponding to figure 2.

discriminability of these textures. Examples of binary textures similar to those of Purks and Richards are shown in Figs. 4-7. The black and white non-encoded versions of these textures are shown in Figs. 1-4 in another paper [2] in this report. Each texture field was formed by generating a string of binary pulses with carefully controlled statistics. Each string was then broken up into shorter strings which were stacked to form the two-dimensional array which served as the texture pattern. Thus the derived textures are only controlled in one dimension. The texture patterns of Figs. 4-7 were processed with the one-dimensional filter described earlier. The large texture differences result in dramatic color differences between the two halves of the image. Examples of this are shown in Figs. 4-6. In Fig. 7 the color difference is present, but is not nearly so pronounced as in the previous example. This is the expected result since the textures are more difficult to discriminate.

A sinusoidal text target was generated and introduced into the color-encoding system. The results are shown in figure 8. The progression of color is exactly as would be expected. A set of filters was also made with circular symmetry for two-dimensional filtering. The filter used had a blue central disc surrounded by a red annular section with a green ring on the outside. Note that this color ordering is not the same as for the one-dimensional filter. A three-bar chart was imaged using the system. The results are shown in figure 9.

It should be noted that the resultant color in these images represents the spatial frequency distribution in a small, localized area of the textures. This area corresponds roughly to the size of the point spread function of one color segment, say the low frequency segment, of the

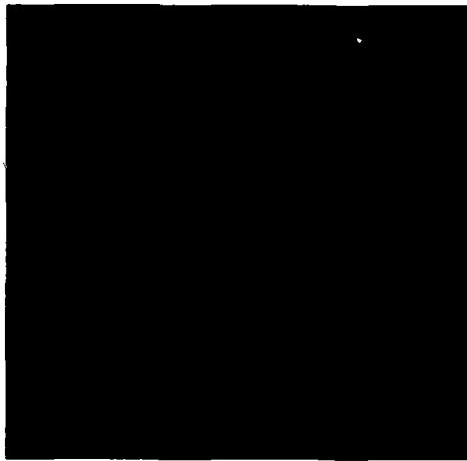


Figure 4

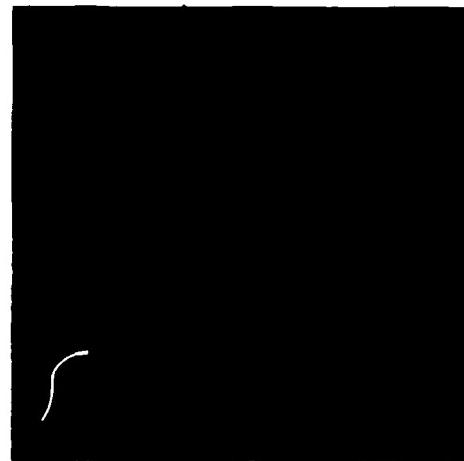


Figure 5

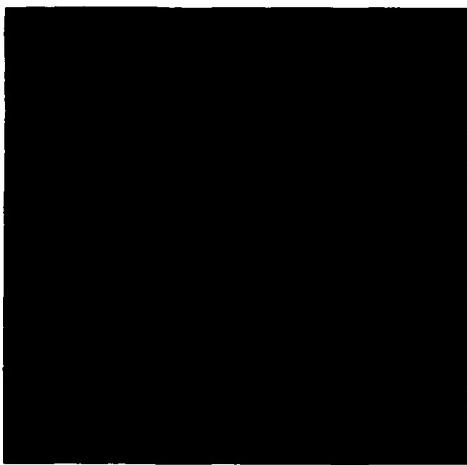


Figure 6

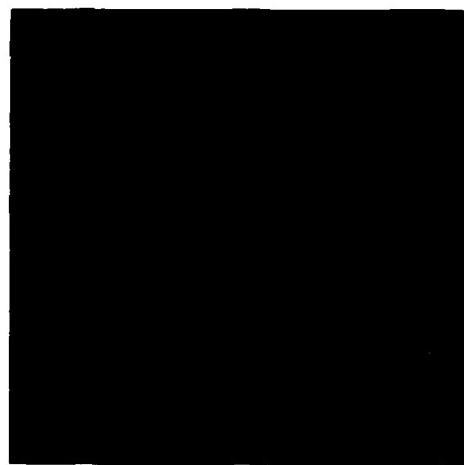


Figure 7

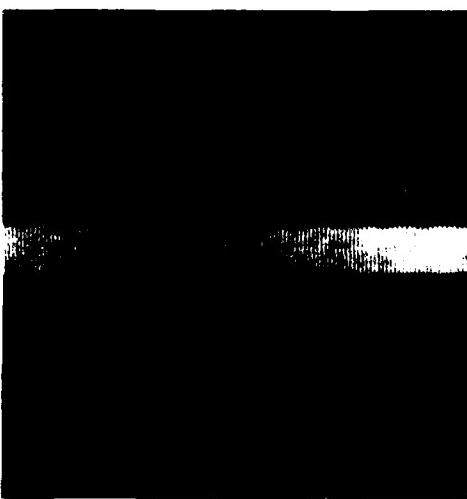


Figure 8

0	1
2	3
3	4
4	5
5	6
6	0
0	1

Figure 9

color filter. If it were desirable to determine the spatial frequency content on a coarser scale, a second incoherent imaging step could be added with the appropriate low-pass filter to effectively average the color over a large area. This might be useful if one were interested more in the spatial frequency distribution over large areas of the image and not in fine details in the image.

Conclusion

We have presented a system that color encodes the local spatial frequency content of an image. The system has been shown capable of color coding certain artificial textures. Such coding could be useful either for making small texture differences more readily detectable for a human observer or it could serve as preprocessor which provides information of spatial frequency content to an electronic or optical processing system.

References

1. H.C. Andrews, A.B. Tescher and R.P. Kruger, IEEE Spectrum, 9, 7, 20 (1972).
2. D.D. Garber, "Generation and Discrimination of One-dimensional Binary Textures," University of Southern California, USCIP Report, Sept. 1978.
3. J.J. Burch, J. Opt. Soc. of Am. 60, 709A (1970).
4. J. Bescos and T.C. Strand, Applied Optics, 17, 2524 (1978), (accepted for publication)
5. S.R. Purks and W. Richards, Opt. Soc. Am., Vol. 67, pp. 765 (1977).

2.4 One-Dimensional Texture Pattern Generation and Discrimination

David D. Garber

Introduction

Texture is a complex image attribute that is difficult to define precisely and has been the subject of much research. The relationship between discrimination of textures by human observers and the mathematical attributes of textures has also been extensively researched. Models have been proposed to allow computer discrimination based on statistical parameters considered in some aspects to be primary texture measures.

Julesz [1] created computer generated patterns with controlled high-order statistical properties. A conclusion drawn from his work is that texture fields differing only in third and higher order statistics cannot be discriminated by a human viewer. Pollack [2] has shown later that textures whose first and second order nearest neighbor probabilities are equal may be discriminated by varying the third order nearest neighbor probabilities. Purks and Richards [3] extended this concept to create texture patterns that differ only in their statistics for four adjacent points. This study indicates that such textures can also be easily discriminated.

However as was pointed out by Pratt [4], the second order probability pairs of the two fields are not constrained to be equal for an arbitrary pixel along an image line. Thus there is some question still as to the

relationships between measured mathematical parameters and human discriminability.

We have studied in detail the mathematical relationships of parameters involved in binary computer-generated one-dimensional texture patterns. Texture patterns in this paper have been generated with specific goals in mind using the mathematical relationships derived. Methods have been developed to control texture statistics for both nearest neighbor and non-nearest neighbor cases and corresponding textures are presented.

Generation Procedure

One-Dimensional binary textures represent the simplest form of texture possible. It is believed that such binary patterns force human observers to utilize primitive visual mechanisms in discrimination. They are not designed to replace or imitate natural textures but are experimentally valuable in deriving concepts concerning texture attributes due to their mathematical non-complexity.

In this experiment, binary sequences with carefully-controlled transition probabilities dependent on the last 4 points were generated and transferred to an image texture file. Each sequence was then broken up into shorter strings which were stacked to form the two-dimensional array which served as the texture pattern. Thus, the derived statistics are only controlled in one dimension.

We can allow the a priori probability of a binary sequence of length N to be defined by $P(V_1, V_2, \dots, V_N)$ where each V_I , $I = 1, N$ is either 0, or 1. As our binary sequence is controlled, in fact, determined, by generation probabilities we need to define a set of parameters

$G(v_1, v_2, \dots, v_N)$ which represents the probability of generating a 0 after the binary sequence v_1, v_2, \dots, v_N . It follows that the probability of generating a 1 after the sequence v_1, v_2, \dots, v_N is $1 - G(v_1, v_2, \dots, v_N)$. Illustration of this commonly-used texture generation method is given by Purks and Richards, however, it should be pointed out that this generation parameters were in many cases constrained to provide equal N-gram statistics, $P(v_1, v_2, \dots, v_N)$. A more general procedure is detailed here.

Mathematics

We are concerned here with a probabilistic approach to generating textures beyond that presented in the Purks and Richards article. This interest is motivated by the desire to generate any pattern according to a set of given probabilities $P(v_1, v_2, \dots, v_N)$ which may be named the N-gram statistics of a specific pattern. We must therefore deal with the relationships that exist between these N-gram statistics and their generation parameters denoted by $G(v_1, v_2, \dots, v_N)$. Examining these relationships and also those between N-gram statistics of different lengths, that is the relationships between $P(v_1, v_2, \dots, v_I)$ and $P(v_1, v_2, \dots, v_J)$ for all I and J, leads us to an understanding of the probabilistic system involved and thereby a method of generating desired texture patterns.

Considering the nature of the experiment, generating random textures, rather than rigorously define a probability function on a sample space, it is just as informative to draw a simple analogy to our generation process and from this draw some basic concepts and conclusions. We recall that $G_1(v_1, v_2, \dots, v_N)$ was defined as the probability of generating a "1" following the sequence (v_1, v_2, \dots, v_N) where each v_i is either a "0" or "1". We also recall that

$$G_0(v_1, v_2, \dots, v_N) = 1 - G_1(v_1, v_2, \dots, v_N).$$

(The modification of the sequence $(v_1, v_2, \dots, v_{N-1})$ to (v_1, v_2, \dots, v_N) from the Purks and Richards paper does not change their validity as these relationships hold for all N.) $G_0(v_1, v_2, \dots, v_N)$ is the probability of generating a "0" following the sequence v_1, v_2, \dots, v_N .

We might regard this generation process to be equivalent to the experiment consisting of tossing a "smart" coin that has a finite memory. In this case, $G_0(v_1, v_2, \dots, v_N)$ might represent the probability of tossing a "heads" given the previous sequence of N tosses was v_1, v_2, \dots, v_N . The resulting string of "1's" and "0's" (0 is the random variable denoting heads, 1 denotes tails) recorded from this experiment is our "texture". We realize immediately that the texture is "determined" by a set of generation parameters $G_0(v_1, v_2, \dots, v_N)$. Using the concept of conditional probability where $P(A/B)$ is the probability of A given B we notice that

$$G_0(v_1, v_2, \dots, v_N) = P(0/v_1, v_2, \dots, v_N)$$

Perhaps the most important concept derived from these generation parameters is that of the finite memory of the system. As is indicated by the notation $G_0(v_1, v_2, \dots, v_N)$, the probability of generating a zero depends on the string of binary values v_1, v_2, \dots, v_N and not those "preceding" v_N . It is thereby suggested that our system has an N-gram memory and we will define such a system N-dimensional. For example, returning to our coin tossing experiment, if we are in a four-dimensional system, the probability of tossing a head depends on the four previous tosses only and all these conditional probabilities are determined by the sixteen parameters $G_0(v_1, v_2, v_3, v_4)$.

With these concepts in mind we return to our initial problem, find these generation parameters $G_0(v_1, v_2, \dots, v_N)$ given the desired probabilities $P(v_1, v_2, \dots, v_N)$ and vice versa. The approach taken by Purks and Richards [3] in finding these N-gram statistics seems to be based on sampling the generated textures. This may be seen by examining the entries in their Table I. The entries seem to correspond to the number of each N-gram counted in the texture generated and the accuracy of such probabilities depends on the law of large numbers. So the true probabilities $P(v_1, v_2, \dots, v_N)$ are only approximated by the output textures and this approximation is poor when the physical size of the textures is small, that is the number of N-grams output is small, the probabilities are close to zero or one, the length of the N-gram string, is large. Therefore it is desirable to compute the exact probabilities given the generation parameters of the system.

Before proceeding further it is useful to prove an identity:

$$P(v_1, v_2, \dots, v_{N-1}) = P(v_1, v_2, \dots, v_{N-1}, 0) + P(v_1, v_2, \dots, v_{N-1}, 1).$$

$$P(v_1, v_2, \dots, v_{N-1}, 1) = P(v_1, v_2, \dots, v_{N-1}) * G_1(v_1, v_2, \dots, v_{N-1})$$

$$\begin{aligned} P(v_1, v_2, \dots, v_{N-1}, 0) &= P(v_1, v_2, \dots, v_{N-1}) * G_0(v_1, v_2, \dots, v_{N-1}) \\ &= P(v_1, v_2, \dots, v_{N-1}) * (1 - G_1(v_1, v_2, \dots, v_{N-1})) \end{aligned}$$

therefore

$$P(v_1, v_2, \dots, v_{N-1}, 0) + P(v_1, v_2, \dots, v_{N-1}, 1) = P(v_1, v_2, \dots, v_{N-1})$$

As a result we have the following three sets of equalities

$$P(v_1, v_2, \dots, v_{N-1}, 0) = P(v_1, v_2, \dots, v_{N-1}) * G_0(v_1, v_2, \dots, v_{N-1}) \quad (1)$$

$$P(v_1, v_2, \dots, v_{N-1}, 1) = P(v_1, v_2, \dots, v_{N-1}) * (1 - G_0(v_1, v_2, \dots, v_{N-1}))$$

$$P(v_1, v_2, \dots, v_{N-1}) = P(v_1, v_2, \dots, v_{N-1}, 0) + P(v_1, v_2, \dots, v_{N-1}, 1) \quad (2)$$

$$G_0(v_1, v_2, \dots, v_{N-1}) = \frac{P(v_1, v_2, \dots, v_{N-1}, 0)}{(P(v_1, v_2, \dots, v_{N-1}, 0) + P(v_1, v_2, \dots, v_{N-1}, 1))} \quad (3)$$

(3) results from (1) and (2) and is essentially a statement of Bayes theorem for our problem. It should be noted that equation (4) of Purks and Richards [3] article is not included in the above as such an equation may be used only when comparing (generating) patterns with different N-gram statistics where the shorter ($n-1$) transition probabilities are fixed. We are producing a more general system. One should also notice that equality (2) is reflected in the entries in Table I of Purks and Richards article (except for column 2 which the author of this paper believes to be in error). Naturally, the numbers in these columns must first be scaled by a common denominator such that the sum of $P(v_1, \dots, v_N)$ is equal to one for any N before this is clearly seen.

Let us then consider the problem of obtaining the generation parameters (G 's) from the probability parameters (P 's). Immediately we come to the conclusion that this is a trivial problem. One might merely use equality (3) to deduce the generation parameters. For example, consider the

problem where we desire to generate a texture with the following 3-gram statistics

$$\begin{array}{ll} P(0,0,0) = 0.0125 & P(1,0,0) = 0.2375 \\ P(0,0,1) = 0.2375 & P(1,0,1) = 0.0125 \\ P(0,1,0) = 0.0125 & P(1,1,0) = 0.2375 \\ P(0,1,1) = 0.2375 & P(1,1,1) = 0.0125 \end{array}$$

Using (3) we obtain

$$G_0(0,0) = 0.05$$

$$G_0(0,1) = 0.05$$

$$G_0(1,0) = 0.95$$

$$G_0(1,1) = 0.95$$

Recall that the $G_1(V_1, V_2)$ are defined also as $G_1(V_1, V_2) = 1 - G_0(V_1, V_2)$. And we could generate textures using generation parameters derived in this way but we will soon find out why this is an incorrect approach. For example, consider the following set of 3-gram statistics.

$$\begin{array}{ll} P(0,0,0) = 0.015 & P(1,0,0) = 0.19 \\ P(0,0,1) = 0.285 & P(1,0,1) = 0.01 \\ P(0,1,0) = 0.01 & P(1,1,0) = 0.285 \\ P(0,1,1) = 0.19 & P(1,1,1) = 0.015 \end{array}$$

Using (3) we obtain the exact same set of generation parameters! We arrive at a contradiction as we can not use the same generation parameters $G_0(V_1, V_2)$ and obtain two different sets of N-gram statistics $P(V_1, V_2, V_3)$. That is, once we have set our generation parameters it follows logically that our texture and its associated $P(V_1, V_2, \dots, V_N)$ are determined for all N.

Observing the equality that we used to obtain these generation parameters (3) we see that the P's do in fact determine the G's but there is no indication that given the

G 's, the P 's will be returned in the generation process. In other words, the mapping from the set of P 's to G 's is not one-to-one. In fact the mappings between the N -gram statistics and generation parameters of the texture generation system are rather complex. We will examine those relationships in more detail.

As was stated above, once the generation parameters are defined, a texture may be generated using those parameters and the N -gram statistics are determined. We also know that once a complete set of N -gram statistics $P(V_1, V_2, \dots, V_N)$ are defined for some N the N -gram statistics $P(V_1, V_2, \dots, V_M)$ may be resolved using (2) for all $M < N$. Again we restate the problem. Given the generation parameters of a system, $G_0(V_1, V_2, \dots, V_N)$, determine the N -gram statistics, $P(V_1, V_2, \dots, V_N)$ of the resulting texture.

The solution to this problem may be found by considering the generation procedure as a discrete Markov process. This approach is readily seen when considering the generation parameters $G_0(V_1, V_2, \dots, V_N)$ as transition probabilities. If we consider a two-dimensional system with $P(0,0), P(0,1), P(1,0)$ and $P(1,1)$ and generation parameters $G(0,0), G(0,1), G(1,0)$ and $G(1,1)$ we may define our system as composed of four possible states $(0,0), (0,1), (1,0)$ and $(1,1)$. If the system is in state i at the K th observation and in state j at the $(K+1)$ th observation then we say that the system has made a transition from state i to state j at the K th stage of the generation process. In our example an observation is taken at each generation of a single new binary value and the state is determined by the values of the last two binary numbers generated. As an example, consider the sequence $0, 1, 1, 0, 0$. We might consider the system to be in the $(0,1)$ state at the start which may represent the K th stage of our generation process then a transition is

made to the (1,1) state at the (K+l)th stage. These transition probabilities are determined by the generation parameters of the system. We also note that our n-dimensional system has 2 possible states. As the transitions from each of these 2 possible states to each of the the 2 possible states is fixed by our generation parameters we may form a transition matrix T whose elements $t(i,j)$ represent the probability of a transition from the ith state to the jth state. If T is the transition matrix of a regular Markov chain, then there is a unique probability vector p which has positive coordinates and satisfies

$$T^T p = p$$

This same vector p may be computed by taking any row of the matrix

$$T^q$$

as q approaches infinity [5]. The vector p represents the vector of steady state probabilities. In our case it contains the desired probabilities $P(v_1, v_2, \dots, v_N)$.

It is important to realize that this theorem holds for regular Markov processes. There is a set of absorbing Markov chains which are formed when any element of the transition matrix is equal to one along the diagonal. This could happen if $G_0(0,0)=1$ for example (a series of 0's would be generated in this case). For the purposes of our discussion we will assume that

$$0 < G_0(v_1, v_2, \dots, v_N) < 1$$

for all v_i . This is a sufficient but not necessary condition for the process to be non-absorbing.

Applying these concepts to a two-dimensional system we obtain the transition matrix

$$\begin{bmatrix} G_0(0,0) & 1-G_0(0,0) & 0 & 0 \\ 0 & 0 & G_0(0,1) & 1-G_0(0,1) \\ G_0(1,0) & 1-G_0(1,0) & 0 & 0 \\ 0 & 0 & G_0(1,1) & 1-G_0(1,1) \end{bmatrix}$$

The first row contains the transition probabilities from state $(0,0)$ to states $(0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$ in that order. The following set of equations results

$$\begin{bmatrix} G_0(0,0)-1 & 0 & G_0(1,0) & 0 \\ 1-G_0(0,0) & -1 & 1-G_0(1,0) & 0 \\ 0 & G_0(0,1) & -1 & G_0(1,1) \\ 0 & 1-G_0(0,1) & 0 & -G_0(1,1) \end{bmatrix} \times \begin{bmatrix} P(0,0) \\ P(0,1) \\ P(1,0) \\ P(1,1) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

As the above system is singular, we may form an equivalent non-singular set by replacing any equation with

$$P(0,0)+P(0,1)+P(1,0)+P(1,1) = 1$$

by using the fact that p is a probability vector. Solving this system gives the desired N-gram statistics $P(V_1, V_2)$.

Examining these generation parameters further we find that the same N-gram statistics may be generated by generation parameters of a different dimension. For example, the following two sets

$$\begin{array}{l|l|l} G_0(0,0) = 0.05 & G_0(0,0,0) = 0.05 & G_0(1,0,0) = 0.05 \\ G_0(0,1) = 0.07 & G_0(0,0,1) = 0.07 & G_0(1,0,1) = 0.07 \\ G_0(1,0) = 0.92 & G_0(0,1,0) = 0.92 & G_0(1,1,0) = 0.92 \\ G_0(1,1) = 0.75 & G_0(0,1,1) = 0.75 & G_0(1,1,1) = 0.75 \end{array}$$

The values, $G_0(V_1, V_2, V_3)$ of the second set indicate that the system is memoryless beyond 2 previous generations that is the probability of generating a zero following a

v_1, v_2, v_3 does not depend on v_1 . We may write

$$G_0(v_1, v_2, v_3) = P(0/v_1, v_2, v_3) = P(0/v_1, v_2) = G_0(v_2, v_3)$$

for all v_2 and v_3 .

It follows that, according to equality (1), the $P(v_1, v_2, \dots, v_N)$ are also determined in our example for $N > 2$ given $P(v_1, v_2)$ and $G_0(v_1, v_2)$. These would be

$$P(v_1, v_2, v_3) = P(v_1, v_2) * G_{v_3}(v_1, v_2)$$

$$P(v_1, v_2, v_3, v_4) = P(v_1, v_2, v_3) * G_{v_4}(v_1, v_2, v_3)$$

$$= P(v_1, v_2, v_3) * G_{v_4}(v_2, v_3) \text{ etc.}$$

We conclude that given any set of generation parameters $G_0(v_1, v_2, \dots, v_N)$ we may form a set of generation parameters $G_0(v_1, v_2, \dots, v_M)$, M greater than or equal to N , according to the rule

$$G_0(v_1, v_2, \dots, v_{M-N}, v_{M-N+1}, \dots, v_M) = G_0(v_{M-N+1}, \dots, v_M)$$

that generate an equivalent set of N -gram statistics and therefore equivalent textures.

Thus far we have solved one half of the problem of our system. Given a set of generation parameters $G_0(v_1, v_2, \dots, v_N)$ we may determine the N -gram statistics of the resulting pattern. We now seek to form a method whereby a set of generation parameters may be found which generates patterns according to an input set of N -gram statistics.

When desiring to generate textures according to a given set of N -gram statistics, $P(v_1, v_2, \dots, v_N)$ we must realize the set of constraints imposed on the set. For example, the sum of the $P(v_1, v_2, \dots, v_N)$ must equal one for all N . Returning to the set of equations used to determine the 2-gram statistics in matrix form we realize that, by adding the first two rows $P(0,1)=P(1,0)$. In fact, by considering the set of equations arising from the set of equations

derived from the generation systems of higher dimensions we find

$$P(v_1, v_2, v_2, v_2, \dots, v_2, v_3) = P(v_3, v_2, v_2, v_2, \dots, v_2, v_1)$$

This implies that many constraints are present on the N-gram statistics. For example, in the 3-dimensional system containing $P(v_1, v_2, v_3)$

$$\begin{aligned} P(0,0,1) &= P(1,0,0) \\ P(0,1,1) &= P(1,1,0) \end{aligned}$$

but also by (2) and the fact that $P(0,1)=P(1,0)$

$$P(0,1,0)+P(0,1,1) = P(1,0,0)+P(1,0,1).$$

Returning to the simple 2-dimensional case we find that the equations arrising from the Markov process give rise to the following set of equations

$$\begin{bmatrix} P(0,0) & 0 & P(1,0) & 0 \\ 0 & P(0,1) & 0 & P(1,1) \end{bmatrix} \times \begin{bmatrix} G_0(0,0) \\ G_0(0,1) \\ G_0(1,0) \\ G_0(1,1) \end{bmatrix} = \begin{bmatrix} P(0,0) \\ P(1,0) \end{bmatrix}$$

A similar matrix is obtained in systems of a higher dimension. This implies that any set of $G_0(v_1, v_2)$ satisfying the above system of equations given that the $P(v_1, v_2)$ satisfy the constraints discussed earlier will generate a texture exhibiting the 2-gram statistics $P(v_1, v_2)$. One might also notice a further constraint on the $P(v_1, v_2)$. As $0 < G_0(v_1, v_2) < 1$, $P(0,1)+P(1,1)>P(1,0)$. The general form of this inequality in an N-dimensional system is

$$P(0, v_1, v_2, \dots, v_N) + P(1, v_1, v_2, \dots, v_N) > P(v_1, v_2, \dots, v_N, 0) \quad (5)$$

Similarly the general form of the equations used to determine the generation parameters from the N-gram statistics is

$$\begin{aligned}
 & P(0, v_1, v_2, \dots, v_{N-1}) * G_0(0, v_1, v_2, \dots, v_{N-1}) \\
 & + P(1, v_1, v_2, \dots, v_{N-1}) * G_0(1, v_1, v_2, \dots, v_{N-1}) \\
 & = P(v_1, v_2, \dots, v_{N-1}, 0)
 \end{aligned} \tag{6}$$

For simplicity this may be rewritten as

$$P_1 * G_1 + P_2 * G_2 = P_3$$

This implies $G_1 = (P_3 - P_2 * G_2) / P_1$ and $G_2 = (P_3 - P_1 * G_1) / P_2$. And so

$$(P_3 - P_2) / P_1 < G_1 < P_3 / P_1$$

and

$$(P_3 - P_1) / P_2 < G_2 < P_3 / P_2$$

But also $0 < G_1 < 1$ and $0 < G_2 < 1$. So finally

$$\text{and } \max(0, (P_3 - P_2) / P_1) < G_1 < \min(1, P_3 / P_1)$$

$$\max(0, (P_3 - P_1) / P_2) < G_2 < \min(1, P_3 / P_2)$$

Thus we can pick any G_1 in the above range and G_2 is then determined or vice versa. In this manner given a set of N-gram statistics we may determine a set of generation parameters which generate a statistical pattern exhibiting those N-gram statistics.

As a last note, consider the p vector whose elements are $P(v_1, v_2, \dots, v_N)$ which does not adhere to the equality and or inequality constraints (4) or (5). It might be desireable to generate a texture whose N-gram statistics are "close to" the input vector p as a texture having the exact statistics of p can not be generated. Equations (4) and (5) define a subspace of possible probability vectors. Therefore an optimal approach to the problem might be to select a newp vector in this subspace such that

$$\text{norm(newp-p)} = \min(\text{norm}(q-p))$$

for all q in the subspace. Using the Euclidean norm, this

becomes a quadratic programming problem and could be approached using those techniques. Such techniques will not be illustrated here. Once the newp vector is found we solve for the generation parameters in the usual way.

In conclusion, a method of determining N-gram statistics from generation parameters using the concept of a Markov chain was determined and an underdetermined system of simple linear equations leading to generation parameters given N-gram statistics was derived. Using this approach to generating texture patterns a large variety of textures may be easily generated and examined using a minimum amount of effort. By these means further understanding of texture perception may be attained.

Further Observations

The above equalities and inequalities provide a full understanding of the texture generation system in probabilistic terms. Still further conclusions can be derived from them. From the above we see more clearly that the generations parameters $G(v_1, v_2, \dots, v_N)$ determine the texture completely and thus define the N-gram statistics $P(v_1, \dots, v_M)$ for all M . Also for a given set of $P(v_1, \dots, v_M)$ there can exist an infinite number of generation parameters which would generate a texture with such statistics or perhaps none at all depending on the relationships derived earlier. Thus textures with equal first, second, third and fourth nearest neighbor probabilities can be generated. Figures 4-6 and their accompanying statistics illustrate this fact.

Parameters thought to be useful in texture discrimination may also be easily developed. For example, joint moments about the mean defined as

$$E[(x_1 - \mu_1)^{r_1} (x_2 - \mu_2)^{r_2} (x_3 - \mu_3)^{r_3} \dots (x_k - \mu_k)^{r_k}]$$

where $\sum_i r_i$ is the order of the moment [6]. The r th moment of x_i is defined as

$$E(x_i^r) = \sum_{x_1} \sum_{x_2} \dots \sum_{x_k} x_i^r f(x_1 \dots x_k)$$

From our binary textures we could define the following parameters.

$$\begin{aligned} \mu &= E\{x_0\} = \sum x_i f(x_i) = \sum x_i P(x_i) = 0 \cdot P(0) + 1 \cdot P(1) \\ &\quad = P(1) \end{aligned}$$

$$\begin{aligned} \sigma^2 &= E\{(x_0 - \mu)^2\} = \sum (x - \mu)^2 f(x) = (0 - P(1))^2 P(0) + (1 - P(1))^2 P(1) \\ &\quad = P(1) - P(1)^2 \end{aligned}$$

$$\begin{aligned} E\{(x_0 - \mu)^3\} &= \sum (x - \mu)^3 f(x) = (0 - P(1))^3 P(0) + (1 - P(1))^3 P(1) \\ &\quad = 2P(1)^3 - 3P(1)^2 + P(1) \end{aligned}$$

$$\alpha = \frac{E\{(x_0 - \mu)(x_1 - \mu)\}}{\sigma^2} = \frac{P(11) - P(1)^2}{P(1) - P(1)^2}$$

$$\theta = \frac{E\{(x_0 - \mu)(x_1 - \mu)(x_2 - \mu)\}}{\sigma^3} = \frac{P(111) - 3P(11) \cdot P(1) + 4P(1)^3}{(P(1) - P(1)^2)^{3/2}}$$

where $P(1)$, $P(11)$, $P(111)$ represent nearest-neighbor (N -gram) statistics although this can be changed to include non-nearest-neighbor statistics thus creating new texture parameters. The above parameters are useful in discrimination therefore only when textures differ in their (3-gram) or shorter statistics.

We can also investigate a method which would allow non-nearest neighbor statistics to be controlled using the relationships developed in the last section. As second-order probabilities have been of interest we might investigate the conditions required to assure equality of

second-order statistics for non-nearest-neighbor statistics. If we denote $G(v_1, v_2, \dots, v_M)$ as our generation parameters and $P(v_1, v_2, \dots, v_N)$ to be their associated N-gram statistics then it may be shown that in order for the 2nd order statistics of one texture to be equal to another for the $(N-1)^{st}$ neighbor distance,

$$\sum_{v_i} P_1(v_1, v_2, \dots, v_M, v_{M+1}, \dots, v_N) = \sum_{v_i} P_2(v_1, v_2, \dots, v_M, v_{M+1}, \dots, v_N)$$

$i=2, N-1$ $i=2, N-1$
for $v_1, v_N \in \{0, 1\}$ where P_1, P_2 represents the N-gram statistics for the first and second texture respectively. It can also be shown that

$$\sum_{v_i} P(v_1, v_2, \dots, v_M, \dots, v_N) = \sum_{v_i} P(v_1, v_2, \dots, v_M) \prod_{k=M+1}^n [v_k + (-1)^{v_k} G(v_{k-M}, v_{k-M+1}, \dots, v_{k-1})] \quad (7)$$

$$i=2, N-1 \quad \text{where } \sum_{v_u} = \sum_{u_j=0}^1$$

$$[v_k + (-1)^{v_k} G(v_{k-M}, v_{k-M+1}, \dots, v_{k-1})]$$

Recall that the N-gram statistics, P , are a function of the generating parameters G . If the 2nd order statistics are to be equal for two statistics regardless of neighbor distance then relationship (7) must hold for all N . But examination of these non-redundant non-linear equations indicates that either the two textures have the exact same generation parameters or the memory of the generation parameters, M , is infinite. Therefore it may be postulated that two binary-one-dimensional textures that are unequal must differ in these 2nd order statistics for some neighbor distance.

Experimental Results

Some of the figures generated using these concepts are shown in Figures 1-12. A set of texture statistics

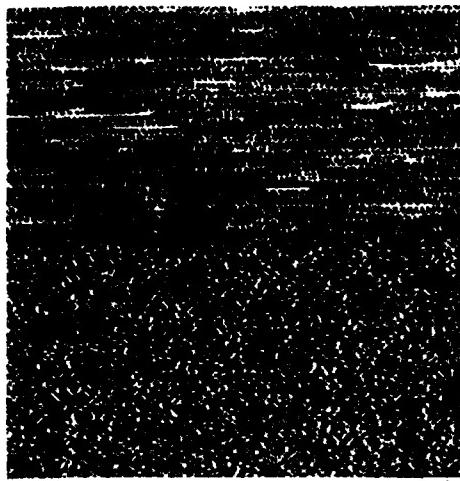


Figure 1

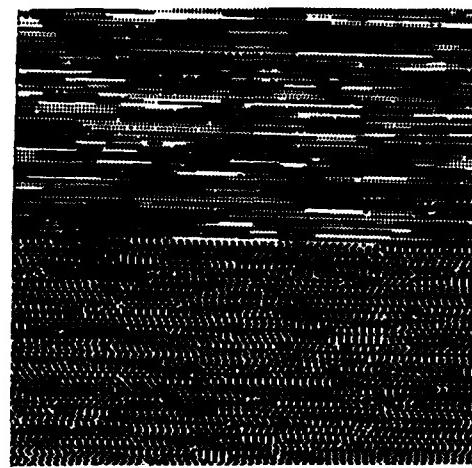


Figure 2

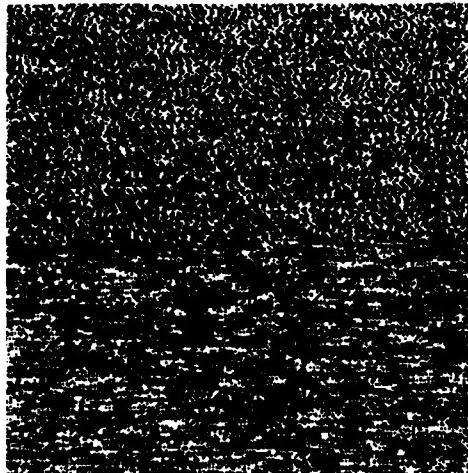


Figure 3

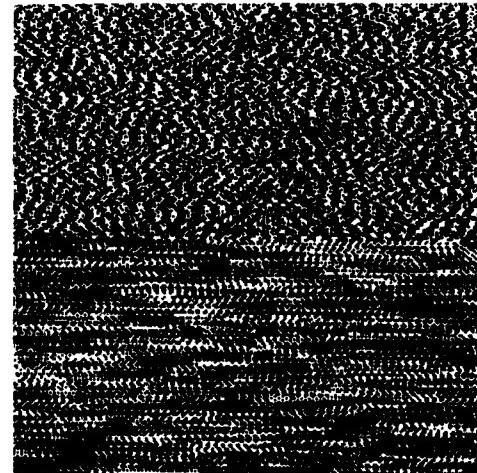


Figure 4

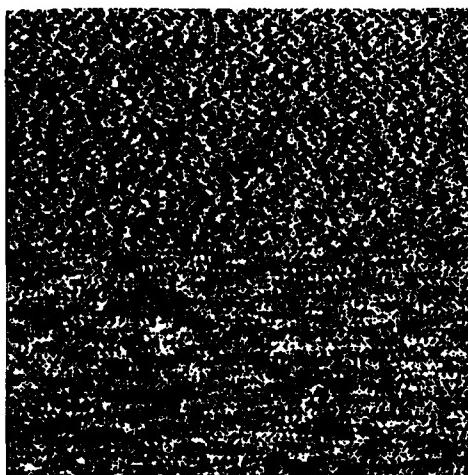


Figure 5

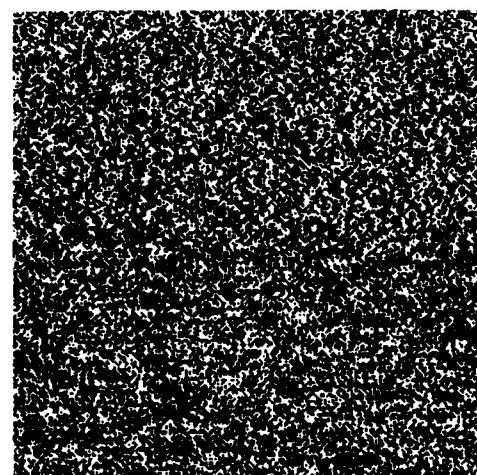


Figure 6

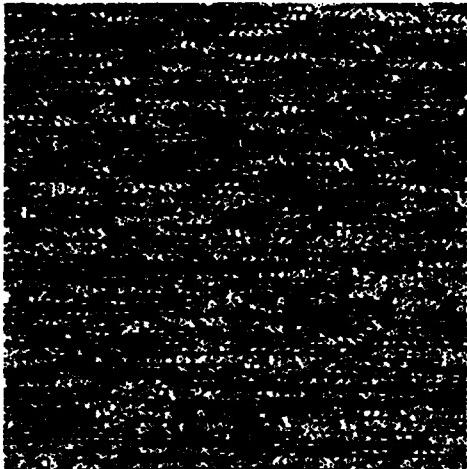


Figure 7

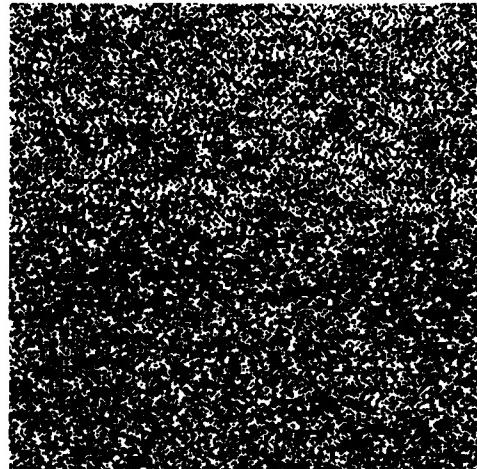


Figure 8

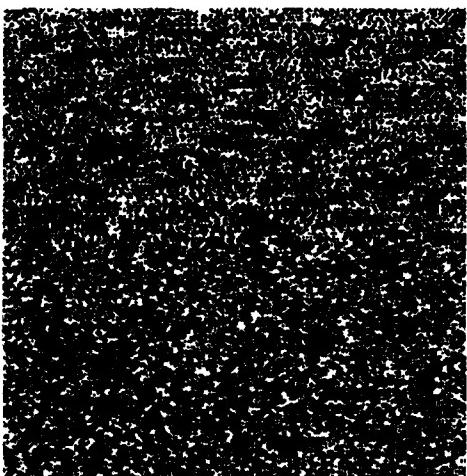


Figure 9

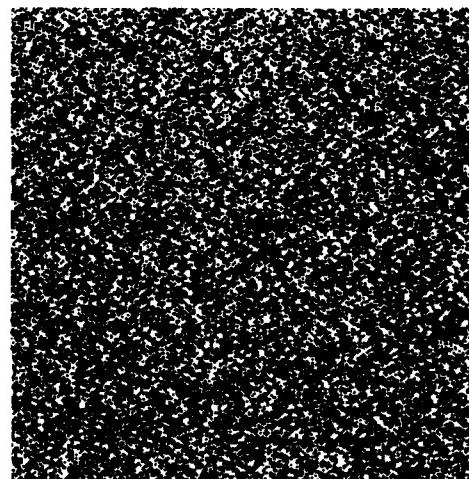


Figure 10

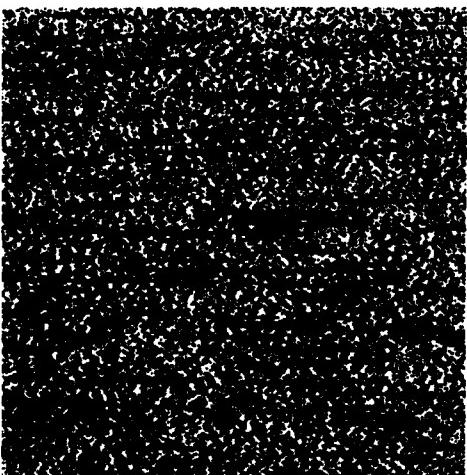


Figure 11

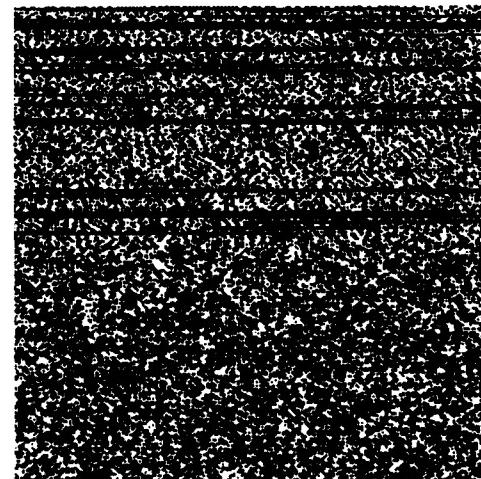


Figure 12

including first, second, third, fourth and fifth-order nearest-neighbor statistics and second-order statistics for non-nearest neighbors was also computed. (These are not included in the report due to their length). The generation parameters from which all other statistics using the ideas presented in the mathematics section may be found are shown in Tables 1-12. These generation parameters represent the $G_0(V_1, V_2, V_3, V_4)$. They are ordered from the top of each column to the bottom as $G_0(0,0,0,0)$, $G_0(0,0,0,1)$, $G_0(0,0,1,0), \dots, G_0(1,1,1,1)$. The left column corresponds to the top texture, the right corresponds to the bottom in each figure.

Figures 1, 2, 3 and 4 are shown in another paper in this report to illustrate the usefulness of a spatial frequency color-encoding system. Figure 1 contains two textures whose 3-gram statistics are equal. Also the various 3-gram statistics are equal within each texture. In Figures 2 and 3 the same is true for only the 2-gram statistics while the 3-gram statistics are shuffled in pairs. Figures 4, 5, and 6 contain textures whose 4-gram statistics are equal between and within but whose 5-gram statistics differ by varying degrees. Figure 7 shows two texture whose 4-grams statistics are equal but neither has equal 4-gram statistics within the texture. Figure 8 shows two textures whose 2-gram statistics for nearest neighbor differ by a fair amount but little visual difference (if any) exists. This texture is one of the more interesting for this reason. Figure 9 contains textures whose 3 and 4 gram statistics differ but whose first, second and third neighbor second-order statistics do not differ. Figure 10 contains two textures whose generation parameters differ in $G_0(0,0,0,0)$ but whose second-order non-nearest neighbor statistics are close to equal and discrimination is not possible. Figure 11 contains two textures whose 5th order

statistics are unequal and whose second order non-nearest neighbor statistics differ, although not severely, but discrimination is not possible. Finally Figure 12 shows two textures whose generation parameters differ almost drastically in $G_0(0,0,0,0)$, the parameters which usually causes a long string of black (0's) to be generated.

Examining the second-order statistics for non-nearest neighbors it is observed that some textures such as those in Figures 4, 5 and 6 have equal statistics except for some certain neighbor distances. That is to say, where we define

$$P(V_1, V_j) = \sum_{V_i} P(V_1, V_2, V_3, \dots, V_j, \dots, V_N)$$

$i \neq 1, j$

to be the $(j-1)$ st neighbor second-order statistic, first ($P(V_1 V_2)$), second ($P(V_1 V_3)$), third...to fifth ($P(V_1 V_5)$) neighbor statistics must be observed to detect differences. In fact, for two general binary textures, second order statistics may be constrained to be equal for up to any arbitrary neighbor, then allowed to differ for further neighbors and visible differences can be noted.

There also seems to be a relationship between the amount of difference exhibited by two textures visually and numerically in their second-order statistics. Without mathematical norms and visual measures as a guide it may be generally concluded that the greater the numerical second-order difference is the greater the visual difference is.

But most importantly these textures indicate that second-order statistics for not only nearest neighbors but also non-nearest neighbors must be used in the discrimination of several one-dimensional binary textures and to what neighbor-distance these measures are to be taken

• 9545	• 5030	• 9792	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 5208	• 5000	• 7500
• 9157	• 9157	• 5000	• 5000	• 7500
• 0933	• 2157	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 9792	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 9157	• 9157	• 5000	• 5000	• 7500
• 0933	• 2157	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 5000	• 5000	• 5000	• 5000	• 7500
• 0455	• 5031	• 5000	• 5000	• 7500

Table 1

Table 2

Table 3

Table 4

Table 5

Table 6

• 8071	• 9725	• 5000	• 8227	• 8548	• 5000
• 7943	• 3423	• 5625	• 5000	• 5000	• 6150
• 8889	• 2722	• 6667	• 5000	• 3498	• 9515
• 4800	• 2532	• 5000	• 3683	• 7401	• 2425
• 6786	• 2321	• 7857	• 2532	• 5000	• 4693
• 9020	• 7420	• 5000	• 7172	• 6785	• 5245
• 5625	• 1263	• 3333	• 5000	• 7778	• 5000
• 1553	• 1463	• 7222	• 5921	• 5000	• 5000
• 9000	• 1000	• 0000	• 5000	• 5000	• 0100
• 1100	• 0000	• 5000	• 2837	• 2111	• 2512
• 8000	• 7000	• 6000	• 5647	• 1399	• 7431
• 2320	• 2882	• 3500	• 5000	• 5670	• 1250
• 4000	• 1002	• 6250	• 3911	• 1967	• 5000
• 3000	• 3157	• 2727	• 7683	• 5000	• 8000
• 0156	• 1250	• 2222	• 4177	• 5381	• 3807
• 9000	• 0000	• 5000	• 5000	• 2954	• 4182

Table 7

Table 8

Table 9

• 1000	• 9000	• 9631	• 9631	• 9948	• 1000
• 7894	• 7894	• 5000	• 1000	• 5000	• 3679
• 6351	• 6351	• 1150	• 1150	• 9000	• 7325
• 0429	• 2429	• 2515	• 4578	• 1224	• 2077
• 6752	• 6474	• 4982	• 4982	• 0675	• 3765
• 0874	• 0974	• 4185	• 3247	• 8776	• 0772
• 1000	• 1000	• 5000	• 9000	• 5000	• 2000
• 2000	• 2042	• 6869	• 6867	• 5755	• 5200
• 0310	• 2037	• 1000	• 1000	• 1000	• 0141
• 9533	• 9333	• 6912	• 9015	• 4480	• 0392
• 8000	• 9000	• 9000	• 9000	• 0916	• 9484
• 1286	• 1226	• 1960	• 2353	• 5000	• 2275
• 3800	• 4300	• 0354	• 0354	• 0061	• 9000
• 4560	• 4640	• 5000	• 8000	• 5000	• 6800
• 6279	• 6272	• 8356	• 7224	• 8776	• 3544
• 9000	• 9000	• 5200	• 5200	• 5200	• 4015

Table 10

Table 11

Table 13

depends on the textures of interest.

Conclusions

The one-dimensional binary patterns generated give rise to some basic concepts concerning textures and their discrimination. First of all they indicate the use of moments and similar statistics is not optimal at least in the nearest-neighbor sense as many textures have equal moment but are visually quite different. However, it should be pointed out that this may only be characteristic of some artificial textures and that moments could serve as good discrimination parameters in many real-world applications. Secondly the results indicate a close relationship between second order non-nearest neighbor statistics and human discrimination. Further, N-gram statistics could provide even more precise texture separation beyond that allowed through use of second-order statistics alone although information content overlaps considerably. Use of these texture measures depends on factors such as discrimination accuracy desired, cost factors for statistics measure and the nature of the textures involved.

If the purpose of texture analysis is to discriminate textures then we must also ask whether our discriminator may be our improvement over the human visual discriminator. That is, do we wish to mimick human analysis or excell it? If the goal is human simulation then the texture study becomes a human visual system and analysis study. If not, the object is texture analysis and discrimination whose results are not necessarily desired or expected to agree with human interpretation of texture. In this case, separation of textures can best be accomplished by statistical measurements on the patterns involved. In either case, success depends on results in the application

of interest. For this reason, no discrimination system will be presented here based on N-gram statistics or non-nearest neighbor second-order statistics.

In future work investigation of two-dimensional textures should be pursued as these correspond more closely to natural scene textures. Still, as is illustrated by the examples here, conclusions drawn from such computer-generated textures may not correspond directly to those drawn from study of natural-scene textures as in most cases computer-generated textures include a far more general class of textures. For this reason, discrimination procedures drawn from computer-generated textures should be more robust and also more mathematically complicated especially if the generation technique allows simulation of natural textures.

References

1. B. Julesz, IRE Transactions on Information Theory, Vol. IT8, 84-92, (1962).
2. I. Pollack, Percept. Psychophys. Vol. 13, pp. 276-280, (1973).
3. S.R. Purks and W. Richards, Journal Optical Society of America, Vol. 67, p. 765-771, (June 1977).
4. W.K. Pratt, et al., "Visual Discrimination of Stochastic Texture Fields," (submitted to IEEE Transactions on Systems, Man and Cybernetics.)
5. Maki and Thompson, Mathematical Models and Applications, Prentice- Hall, New Jersey, (1973).

6. Beyer, Handbook of Tables for Probability and Statistics, The Chemical Rubber Company, Cleveland, (1974).

2.5 Experiments in Natural Texture Description

Keith E. Price and Ramakant Nevatia

While exploring aerial image analysis we undertook several experiments in the use of textural information in the analysis of machine segmented images. When looking at the images (e.g., Fig. 1) it is clear that many areas are characterized by fairly uniform textural patterns. The experiments presented here have not yet provided conclusive results, and do not represent a complete exploration of the texture problem domain, thus no hard conclusions will be drawn, but it was felt that the results should be presented at this time.

We have started with a high resolution aerial image (Fig. 1) (2048 x 2048 points, 3 colors - 8 bits each) and have segmented the image by a machine segmentation method [1] (results in Fig. 2). This initial segmentation locates many of the large areas of the scene, but further analysis will involve recognition of these areas and possible subdivision of the regions. Our interest is in developing texture measures useful for segmentation and recognition of specific regions, such as urban or forest areas, perhaps guided by a priori knowledge of the scene.

From the segmented scenes, windows (64 x 64 or 32 x 32) are selected so that the entire window falls within one segment (a few do overlap 2 different regions) no other selection criteria is applied, except to get a

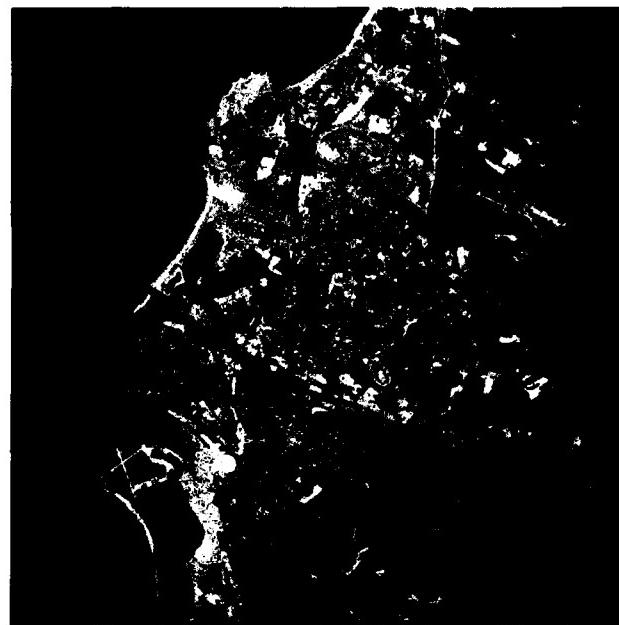


Fig. 1. Green Image of San Diego Area



Fig. 2. Initial Segmentation

representative sample of the type of windows which may occur (in the actual display of results, similar windows are grouped for a better comparison of the results).

Many textural measures have been proposed in the past [2-4] (here limited to statistical rather than structural analysis), therefore one can use some of their results indicating what measures may be useful. Several general techniques will be applied to the original windows and to the windows resulting after various processing steps (some of the operations are included more for completeness than an expectation of useful results).

Among the statistical measures which have been discussed, and used, are analysis of the discrete Fourier transform [6], analysis of generalized gray-level co-occurrence matrices [2], and analysis of the edges (or micro-edges) in a subwindow [5]. We are not interested in finding one texture measure which will distinguish between all regions (this would be the ultimate, but extremely difficult problem) but in finding a texture measure to use in conjunction with many other features of the region [7].

The initial goal of these experiments was to easily distinguish between the regular street pattern of the urban areas and other patterns in the rural (in this case hill sides covered with brush, other cases would include "farm" patterns), water and other regions. This means that we have concentrated on textural measures which give some indication of regular patterns.

Since we have not yet performed detailed analysis on the results, we will just present some of the output and discuss how it was derived and how the results may be used.

Fig. 3 shows the subwindows which were used. There are a total of 64 windows, each is 64 x 64 points. Generally urban areas are in the top left, rural areas in the top right and the bottom half has a mixed collection of windows. These are all from the green image (the original was scanned in color).

Fig. 4 is the magnitude of the Fourier transform [6] for each window. Some structure is visible in the transforms of some of the urban areas, but the same structure may appear in non-urban regions also.

Fig. 5 is the output of an edge detector (magnitude) [8]. Fig. 6 is the Fourier transform, Fig. 7 is the edge directions (8 possible values) and Fig. 8 in its Fourier transform. These seem to be "noise" images, but the structure is very evident in the direction images.

Fig. 9 is the suppressed edge data. Non maximal points (along rows or columns) have been set to 0, therefore edges in the image are represented by only one line. The origin of this suppression method is unknown but is probably due to Rosenfeld. Fig. 10 is its Fourier transform which exhibits more high frequency components than the others, but has stranger patterns when there are few edge elements in the window.

The suppressed data was used for other analysis - number of edges (total number for each window is given in Fig. 11) and binary co-occurrence analysis. Because we are only using binary data the co-occurrence computation can be simplified and only a few measures on the co-occurrence matrix seem potentially useful. The numbers shown in Fig. 12 are for: $(\sum \text{edge AND edge}) / (\sum \text{edge OR edge})$ and $(\sum \text{edge AND edge}) / (\text{all possible pairs})$ for separation = 1, 2, 4, and 8,



Fig. 3. Texture areas for study

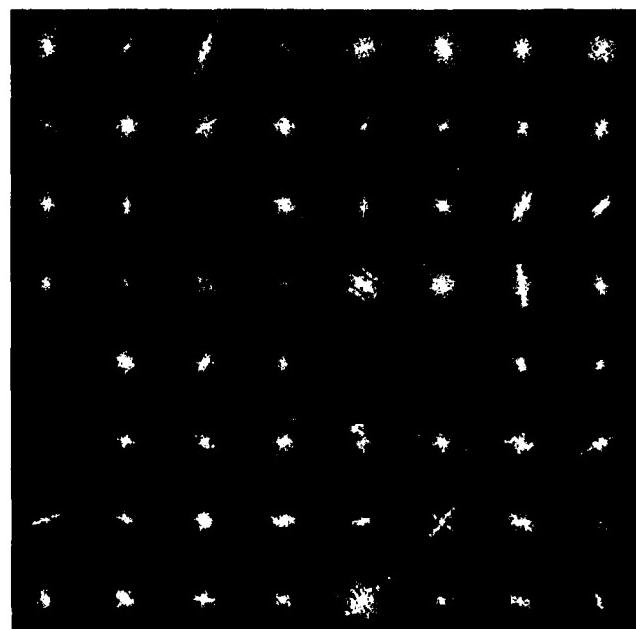


Fig. 4. Fourier Transforms of image window in Fig. 3



Fig. 5. Edge operator output (Magnitude) of window in Fig. 3

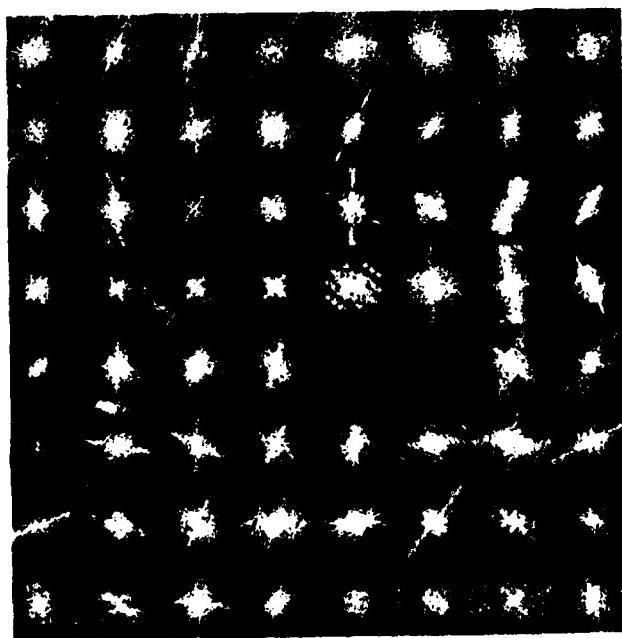


Fig. 6. Fourier Transform of edge windows in Fig. 5



Fig. 7. Edge Detectors output (Directions) of windows in Fig. 3



Fig. 8. Fourier Transform of directions window in Fig. 7

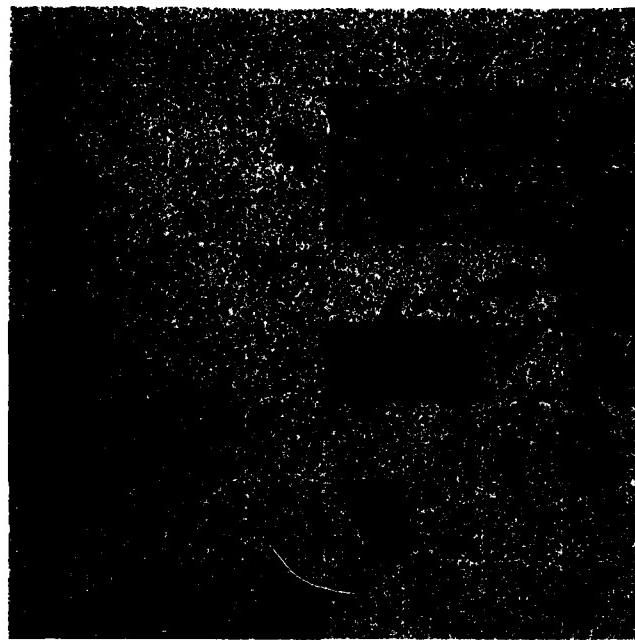


Fig. 9. Non Maximal Supressed Edges from Fig. 5

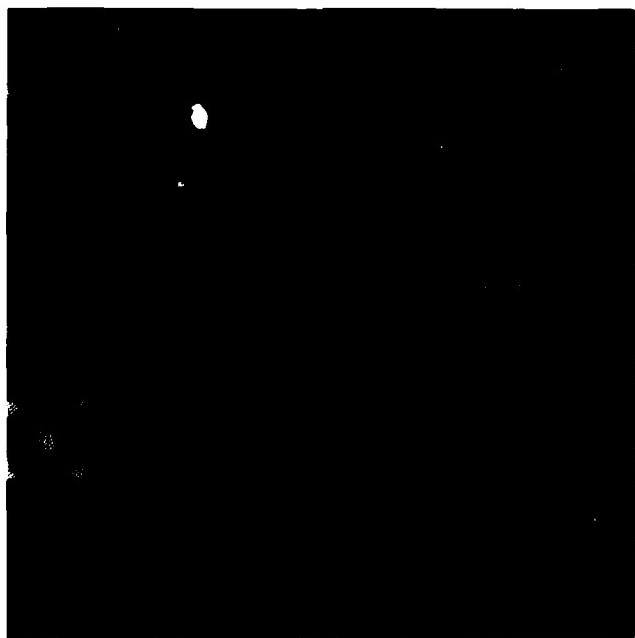


Fig. 10. Fourier Transforming of Binary suppressed edges

1475	1542	1424	1542	1545	1312	1383	1322
1567	1529	1577	<u>1219</u>	374	395	380	516
<u>794</u>	1582	1552	1597	481	558	615	504
1312	1240	1546	1223	1458	1295	1031	169
31	1260	1156	1330	0	0	1215	860
8	930	854	1323	1319	1433	1115	1078
1457	1416	1219	1318	506	1506	1570	1612
1245	1381	1207	753	1669	1617	1667	1478

Figure 11. Number of Edges in the Subwindow

ΔI	ΔJ	0	1	2	4	8	ΔI	ΔJ	0	1	2	4	8
0	0	18,11	22,13	22,13	21,13	21,13	0	0	17,9	17,9	20,10	20,10	19,10
1	21,12	24,14					1	17,9	17,9				
2	25,14		20,12				2	15,8	15,8	20,10			
4	25,14			23,13			4	16,9	16,9		16,8		
8	24,13				20,11		8	16,9	16,9		15,7		

Urban area window 1,1

Window 5,2

ΔI	ΔJ	0	1	2	4	8	ΔI	ΔJ	0	1	2	4	8
0	0	16,11	26,16	24,15	24,15	24,15	0	0	15,5	15,5	17,6	17,6	11,4
1	25,14	27,16					1	9,5	9,5	13,5			
2	29,17		24,15				2	7,4	7,4	13,5			
4	28,17			22,13			4	7,4	7,4	13,5			
8	28,16				21,13		8	8,4	8,4		10,4		

Urban area window 3,4

Window 7,3

Fig. 12. Cooccurrence Percentages. First number is 11 as a percentage of 11+10+01, second is 11 as percentage of 11+10+01+00. The ΔI & ΔJ values are given at the side & top of the display, no entry indicates cooccurrence was not computed for this combination. The window position is row, col.

AD-A134 989

IMAGE UNDERSTANDING RESEARCH(U) UNIVERSITY OF SOUTHERN
CALIFORNIA LOS ANGELES IMAGE PROCESSING INST

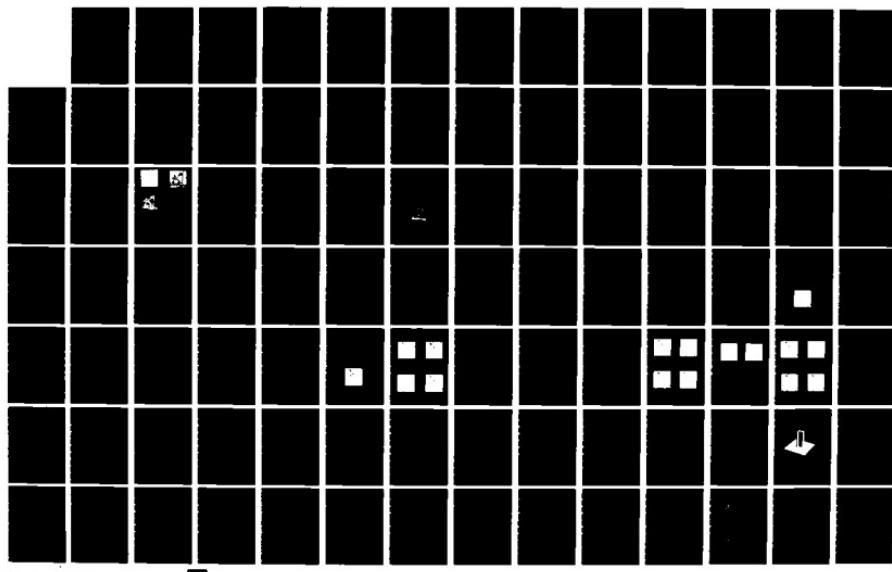
2/4

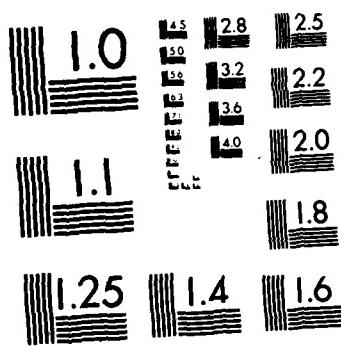
H C ANDREWS 30 SEP 78 USCIP1-840 F33615-76-C-1203

F/G 20/6

NL

UNCLASSIFIED





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ΔI	ΔJ	0	1	2	4	8	16
0	0	15, 3	10, 2	7, 1	6, 1	0	19, 12
1	3, 1	17, 3				1	23, 14
2	2, 1		12, 2			2	23, 14
4	1, 0			7, 1		4	23, 13
8	1, 1				4, 1	8	22, 13

Rural area window 2, 6

ΔI	ΔJ	0	1	2	4	8	16
0	0	18, 4	11, 2	8, 2	8, 2	0	15, 10
1	3, 1	26, 5				1	25, 14
2	3, 1		12, 3			2	30, 17
4	2, 1			7, 2		4	29, 17
8	2, 1				7, 2	8	29, 17

Window 1, 5

Window 6, 8

Fig. 12. (Continued)

pixels and angle = 0° , 45° and 90° .

These results are from preliminary experiments which are being carried out to aid other research in general analysis of aerial images. The initial results seem to indicate that the number of threshold edge (Fig. 11) would distinguish between the rural and urban areas better than the others presented - but there is not sufficient data to make any general statement yet.

References

1. R. Ohlander, K. Price and R. Reddy, "Picture Segmentation Using Recursive Region Splitting Method," to appear Computer Graphics and Image Processing.
2. R.M. Haralick, K. Shanmegar, and I. Dinstein, "Textural Features for Image Classification," IEEE TRANS SMC-3, 1973, pp. 610-621.
3. R. Bajcsy, "Computer Identification of Visual Surfaces," Computer Graphics and Image Processing-2, Oct. 1973, pp. 118-130.
4. J. Weszka, A. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification," IEEE Trans SMC-6, April, 1976.
5. A. Rosenfeld, A. Kak, Digital Picture Processing, Academic Press, New York, 1976.
6. W. Pratt, Digital Image Processing, Wiley: New York, 1978.
7. K. Price, R. Nevatia, "Locating Structures in Aerial

Images," Proc. 4 Intl. Joint Conf. on Pattern Recognition, Kyoto, Japan, Nov. 1978.

8. P. Chuan, "Development of Edge Detectors for the Extraction of Linear Segments," USCIP Report 770, Sept. 1977, 37-50.

2.6 Representation and Acquisition of High-Level Image Descriptions

Keith E. Price

In a high level image analysis system the image description facility becomes very important and influences decisions on how the analysis should be performed. The ideas behind the description are the same as many others have had for representing general high level image descriptions, and the major differences would be in the details of the implementation or the actual language used.

These description facilities are used for many analysis operations and thus must be general, extendable, and capable of being stored as text files. The descriptions are initially generated by an automatic segmentation procedure (or a combination of 2 segmentation procedures, one for region like objects and one for line-like objects) or the model description system (described later in this paper). The information is then augmented by feature extraction systems to produce a general feature based symbolic description of the image. The descriptions are also used for object location [1] and image matching systems [2], both for description of the input images and for communications of the results to other programs.

The language being used is SAIL [3] which provided certain language constructs that determined the actual implementation. SAIL is originally based on Algol-60 with many added features. The important additions (for this discussion) are lists and relational triples. The lists are an ordered set of elements (of arbitrary length) so that an element can be referred to by where it occurs in the list (i.e. the 17th element). Relational triples are a method of storing relations between elements. The 3 parts are referred to as property, object, and value (i.e. property OF object IS value, or color OF box IS red, part OF house IS door, intensity OF box IS 121). One of the important features of this relational storage is the ease in programming searches of the relations. For example a list with all the neighbors of a region would be generated by: Neighbors OF Region. This expression can be assigned to a list variable or used as a list expression depending on which is needed. This brief outline of features of SAIL is intended to give enough background so that the later descriptions of the representation will be clear.

The image is represented as a list of elements (regions, lines, objects, group of objects, and partially segmented regions). The properties of the elements and relations between them are stored as relational triples. The list is used primarily for communication to other programs or for separation of several images and would not be necessary if only one image were ever used and there was no need to save the results of the program. The region numbers (i.e. the position in the list) are used to describe relations between regions in the text file representation and to store and refer to information about elements in other images. With the relations between regions, the representation can be expressed as a graph structure with the elements as nodes and relation as arcs.

Each property that is to be saved must be entered in a list of properties. There are many which are available initially but others can be generated and saved at any time. The type of values is varied so that a property type must also be indicated. Properties may be in the form of character strings, integers, real numbers, arrays, and other regions. Additionally the values in integers may be packed 1, 2, or 3 values per word. The property type is important for reading and writing the description file and for accessing the property values. Because of the relational triples, it is possible to allocate the space for features and relations only when needed rather than having a fixed size structure where the existing properties are filled in. The total size is important since an image can easily have 100-200 elements (i.e., in large scale aerial images with 4 million pixels, 200 elements would be a coarse segmentation), and 2 or 3 image descriptions may be needed at one time. Also many feature values or relations may not exist or may not be needed for many elements. Fig. 1 gives an example of 2 regions as they may be printed for the user, in this case other regions are referred to by their region number.

This description system is used both for images and user specified models. The descriptions for an image can be easily derived, but it could be very tedious for a user to specify all the information in the model description. We have developed a program which can facilitate this process. The program was initially designed to interface a naive user to several image analysis programs by asking a sequence of questions where the answer would determine the next questions and the actions by the program.

The set of questions and actions is read into the program at the beginning and can be changed according to the

[15]	ISLAND-2					
	NEIGHBOR	OF	15	IS	3	
	ANCESTORS	OF	15	IS	13 7	
	RED	OF	15	IS	190.00	STD.= 10.00
	GREEN	OF	15	IS	175.00	STD.= 10.00
	BLUE	OF	15	IS	185.00	STD.= 10.00
	TORIENT	OF	15	IS	-.95	
	TSHHTWR	OF	15	IS	.40	
	TFRACT	OF	15	IS	1.20	
	TSQFRACT	OF	15	IS	.30	
	TOLEFT	OF	15	IS	2	
	TABOVE	OF	15	IS	19	
	TNEARBY	OF	15	IS	16 14	
[16]	URBAN-AREA					
	HUE	OF	16	IS	3191 DEGREES	STD.=100
	SAT	OF	16	IS	.0566960	STD.= .0977517
	INTENSITY	OF	16	IS	183.30	STD.= 10.00
	NEIGHBOR	OF	16	IS	18 19 21	
	ANCESTORS	OF	16	IS	7	
	RED	OF	16	IS	190.00	STD.= 10.00
	GREEN	OF	16	IS	175.00	STD.= 10.00
	BLUE	OF	16	IS	185.00	STD.= 10.00
	YINTENSITY	OF	16	IS	307.60	STD.= 10.00
	IINTENSITY	OF	16	IS	265.60	STD.= 10.00
	QINTENSITY	OF	16	IS	268.00	STD.= 10.00
	TCOUNT	OF	16	IS	390 -98304	
	TBDRLEN	OF	16	IS	2000 0	
	TORIENT	OF	16	IS	-.50	
	TSHHTWR	OF	16	IS	.60	
	TFRACT	OF	16	IS	.20	
	TSQFRACT	OF	16	IS	.10	
	TOLEFT	OF	16	IS	2	
	ILOCV	OF	16	IS	1 180	
	JLOCV	OF	16	IS	45 250	
	TBELOW	OF	16	IS	17	
	TNEARBY	OF	16	IS	3 14 15 19	
	REGINNEX	OF	16	IS	-234 47	
	REGSECNEX	OF	16	IS	-1100 51	

Figure 1. Two sample region definitions showing several of the possible data types.

actual task. Rather than limiting the actions to a small set of specific operations, the actions can be any command level (program command level, not operating system command here) operation. Operations at command level are very general with easy access to variables to use as arguments to commands or to return values of commands (also commands which return values can be used as arguments to other commands). There are also commands for controlling other running programs (using features of the TENEX operating system).

The questions are structured in the following way: Each question has an ID by which it is referred to in the text file representation and by the user in all references to questions. Each question also has the text string that is printed out when this question is asked. The question may contain any variables accessible at command level so that the question can be tailored to the current state of the program description (e.g. to refer to objects or relations by name). There is a help entry to provide some guidance on the meaning of the answers (the list of valid responses is obtained directly from the internal representation). A conditional is also included-if it is true the question need not be asked. The conditional, in the case of acquiring model descriptions, is a check of whether the feature has already been defined. This conditional expression may use the command level operations and variables. Finally there is a list of valid answers to the question with special indicators for arbitrary numeric and string responses (string response require a confirmation from the user). Each answer has two entries: a list of the questions to ask next when this answer is given, and the action to take (i.e., the sequence of commands to execute, such as a command to add a feature value or a relation between regions, or send a message to another program). Fig. 2

```
REDQ:What is the RED value of !FIRREG!
ANS:#  
    NQU:  
    PRO:ALPHA_XWORD(MULT(NUMBER,10),100)\DWRITE("RED",FIRREG,DNEW(ALPHA),2)  
ANS:NONE  
    NQU:  
WHY:Why not.  
HELP:This is the RED color parameter, the value is a number  
COND:PROPTEST("RED",FIRREG,2)  
FEATYP:What is the feature value type  
ANS:1  
    NQU:FEASTY  
ANS:2  
    NQU:FEAITY  
ANS:4  
    NQU:FEARTY  
ANS:8  
    NQU:FEAGTY  
ANS:STRING  
    NQU:FEASTY  
ANS:INTEGER  
    NQU:FEAITY  
ANS:REAL  
    NQU:FEARTY  
ANS:REGION  
    NQU:FEAGTY  
WHY:  
HELP:Numeric feature type  
COND:  
FEARTY:What is the feature value (real)  
ANS:#  
    NQU:  
    PRO:DADD(FTYP,FIRREG,DNEW(NUMREAL),4)  
ANS:NONE  
    NQU:  
WHY:  
HELP:  
COND:
```

Figure 2. Three sample questions from the set used to describe image models.

```
@<ZPRICE>PIC
>>QUEST
What is the QUESTION file name: QUERY.MODEL
Describe a region, a model or an object? MOD
Do you want to use an old or new model? OLD SAN-MODEL
SAN-MODEL OK (Y or N) [Y]
+++++
Describe a region, a model or an object? REG
Which region do you want to use? ISLAND-2
ISLAND-2 OK (Y or N) [Y]
Do you wish feature values or relations of ISLAND-2? FEATU
What is the intensity value of ISLAND-2? RED 190 GREEN 175 BLUE 185 NO
What is the size of ISLAND-2? NO
What is the I location of ISLAND-2? NO
What is the length of ISLAND-2's border? NO
What is the orientation of ISLAND-2? -0.95
What is the height/width ratio of ISLAND-2? 0.4
What is the fractional fill of ISLAND-2? 1.2
What is the fractional fill of the MBR of ISLAND-2? 0.3
Do you wish feature values or relations of ISLAND-2? REL
Which relation do you want for ISLAND-2? NEARBY ISLAND-1 NONE
ISLAND-1 OK (Y or N) [Y]
Which relation do you want for ISLAND-2? NONE
Do you wish feature values or relations of ISLAND-2? NONE
Finished with the region Which region do you want to use? NONE
```

Figure 3. Sample Question/Answer session to define Island-2.

shows 3 sample questions in the format used to save the question in a text file.

The question interpreter also has commands for editing the questions. Questions can be added or deleted and individual fields can be changed. For all these commands the questions are referred to by the ID name. The ID name (and indeed the answers to all questions) can be abbreviated to the minimum unique string. Fig. 3 illustrates a short sequence of questions and responses which result in the description of Island-2 (which may be seen in Fig. 1).

References

1. R. Nevatia, and K. Price, "Locating Structures in Aerial Images," Prof. 4th Intl. Joint. Conf. on Pattern Recognition, Kyoto, Japan, Nov. 1978.
2. K. Price and R. Reddy, "Matching Segmentation of Images", IEEE TRAMI, to appear.
3. J.F. Reiser, ed., SAIL, Stanford AI Laboratory Memo AIM-289, Stanford University, Stanford, Calif., August 1976.

2.7 A Proposed Class of Picture Operators

Kenneth I. Laws

This paper proposes a new class of local operators for processing textured images. The operators are based on a window segmented in much the same way that entire images are now segmented. The segments are analyzed and some of their properties are assigned to pixels within the window. The pixels thus take on values or feature vectors which are more

representative of their image regions than were the original values.

Advantages of local operations are rapid processing, independence from unrelated scene components, and mimicing of biological vision systems. The chief disadvantage is lack of global perspective. However global information can be introduced at a later stage after local processing has identified region seeds and other structural properties.

Segmented windows can be used anywhere that unsegmented windows or convolution masks are now being used. Additional power and flexibility are made available by the region knowledge. Applications of this power are suggested in the following sections.

Noise Cleaning

Median filtering is a successful method of removing salt and pepper noise. It is based on the premise that more than half of the pixels in a window will belong to the same region as the center pixel. The center pixel can thus be replaced by the window median with very little degradation in image structure.

I propose an extension which may work well even in textured or colored images. Begin with a large window, say 7x7, and segment it by any of the common image segmentation methods. There may be only one region present or there may be several. Assign a new value or feature vector to the window's center pixel based solely on the statistics of the region to which it belongs. This value may be the median or average of the region, or perhaps a linearly weighted prediction. In any case it will be uncontaminated by other regions within the window. This segmentation filter should

effect considerable edge sharpening since it forces each edge pixel to be adjusted toward one region or the other.

Figure 1 illustrates the use of a segmentation filter on an image with two distinct regions. The filter assigns to the center pixel the median of its region. It is assumed that each window is segmented perfectly. Results are also shown for unsegmented mean and median filters. In both the 3x3 and 5x5 cases the segmented filter produces a sharper boundary.

Texture Measurement

Many researchers have used windows to gather texture statistics. This makes sense only if the window contains a single texture. The segmented window allows statistics to be calculated over just the region to which the center pixel belongs. These statistics (e.g. mean, variance, co-occurrence) are then assigned to the center pixel as texture features. This is a multivariate application of the previously discussed segmentation filter.

Region Seed Identification

Some algorithms seek "quiet" windows over which to compute texture measures. This edge avoidance not only simplifies texture computation, but also locates uniform areas which make better region seeds than areas near region boundaries. Identification of uniform areas is trivial with segmented windows: just use the windows which consist of single regions. These windows can be further screened if it is desired to have only one seed from each large region.

The segmented window allows seeds to be found for smaller regions as well. For each new window region a

2	3	4	6	7	8				
2	3	3	6	7	7	3.0	4.3	5.8	7.0
3	3	4	7	8	7	2.8	4.2	5.8	7.1
2	2	3	7	7	8	2.9	4.2	5.9	7.2
3	3	3	6	8	7	2.8	4.2	5.9	7.3
2	3	4	7	8	8	3.0	4.2	5.9	7.1
3	2	4	6	7	7	2.9	4.3	6.0	7.2
2	3	3	7	8	7				

a) Original image

b) 3x3 mean

3	4	6	7
3	3	7	7
3	3	7	7
3	3	7	7
3	4	6	7
3	4	7	7

c) 3x3 median

3	3	7	7
3	3	7	7
3	3	7	7
3	3	7	7
3	3	7	7

d) 3x3 segmented median

3	6
3	7
4	7
3	7

e) 5x5 median

3	7
3	7
3	7
3	7

f) 5x5 segmented median

Figure 1. Comparison of Filter Types

representative pixel or feature vector can be stored. Seeds can then be chosen by spatial or cluster analysis of the stored data.

Edge Detection

The segmented window can be adapted to identification of edge elements within an image. In a segmented window the boundaries between regions have been located. These boundaries are less certain toward the edges of the window, but only the information at the center is to be kept as the operator output.

The simplest edge measures assume that edges pass only between pixels and only at particular orientations. It is easy to keep track of such edges, say above and/or to the right of the center pixel. (For this application it would be better to use an even-length window centered over an inter-pixel gap.) After one image pass every inter-pixel gap will be classified as edge or non-edge. Edge strength and direction can also be recorded.

A more sophisticated edge detector would allow for region boundaries passing through pixels. An edge element would be fitted through the center pixel if a region boundary was found near it. Region statistics on each side of the boundary could be used to measure edge strength. Sharp corners or trihedral vertices could also be identified in this manner. It is difficult to store and work with such high resolution edge data, but methods have been developed [1].

Edge following is also made easier by segmented windows. To extend an edge, simply create the next window along the edge direction. It contains sufficient

information to determine the behavior of the line at the next pixel. The fact that a line is being followed may even help in properly segmenting the window.

Some algorithms, notably Yakimovsky's [2], require a measure of edge strength between every pair of adjacent pixels. The segmented window increases the information available for measuring edge element strength. It is possible, for instance, to include measures of the straightness or average strength of the boundary between two regions. Region statistics can also be used in hypothesis testing and strength measurement.

Implementation

Segmenting a window differs from segmenting an entire image in that fewer pixels are available. This speeds computation but reduces the amount of information available. If the window is large enough there will seldom be an error affecting the center pixel. Even these errors should have little effect if the window operation is only a preliminary to global image segmentation.

The optimal window size will depend on the application and the image, and may even vary within an image. One method of choosing a window size is discussed by Deguchi and Morishita [3]. For ad hoc operators it is probably necessary to try several sizes to see which works best.

There are many methods of segmenting images. It is not known which is best for window segmentation, but some opinions can be offered. The key problems seem to be identifying the number of window regions and finding good seeds from which to grow them. One-pass algorithms are probably crude, but may be adequate for some purposes.

Pyramid or planning methods are of doubtful use since the window resolution is so coarse. Non-spatial methods such as histogram segmentation [4] or cluster analysis should be sufficiently fast, and the window size limits the likelihood of disconnected regions.

Global information can be used to speed up segmentation even at this early stage. One way is to use global seeds to initiate cluster formation. The seed textures can be taken from an application-dependent database or simply from region textures found in previous windows. For real-time or filmstrip analysis the seeds can be retained from one frame to the next.

Another speedup technique is to retain segmentation data when the window is shifted. Only the pixels in the new row or column need to be classified. Unfortunately this reduces the amount of information contributed by each window. It is adviseable to reclassify each pixel using the updated region statistics.

Processing time can also be saved by using less window overlap. One could, for instance, use 6x6 windows shifted two spaces, and keep the center 2x2 element. This would require complete restructuring of the operational algorithms.

If the segmented window proves sufficiently valuable, it could be implemented in real-time hardware. This would greatly increase the computational power of smart sensors and image processing equipment.

References

1. Nevatia, Ramakant, "Locating Object Boundaries in

Textured Environments", IEEE Trans. on Computers, Vol. C-25, No. 11, Nov. 1976, pp. 1170-1175.

2. Yakimovsky, Yoram, "Boundary and Object Detection in Real World Images," J. ACM, Vol. 23, No. 4, Oct. 1976, pp. 599-618.
3. Deguchi, Koichiro and Iwao Morishita, "Texture Characterization and Texture-Based Image Partitioning Using Two-Dimensional Linear Estimation Techniques," IEEE Trans. on Computers, Vol. C-27, No. 8, August 1978, pp. 739-745.
4. Ohlander, R., K. Price, and R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," Computer Graphics and Image Processing, (in press).

2.8 An Edge Detection, Linking and Line Finding Program

Ramakant Nevatia and K. Ramesh Babu

The goals of this work are to develop low level line finding programs adequate for processing complex images of interest for higher level image understanding tasks. In spite of the large amount of previous research in this area, no algorithms suitable for complex imagery are apparent. In particular we found the widely used Hueckel operator to be deficient for images with fine detail and texture. In this section we describe an edge detection and line finding technique with superior performance. These algorithms are presented as pragmatic solutions to the low level problems of image understanding with little discussion of their optimality or novelty.

The described algorithms exhibit good performance on a variety of images and are already being used by Hughes Research Laboratories on an independent program and being considered for use by Tom Binford's group at Stanford. These algorithms are largely local in nature and can be applied to large images without difficulties of storage (but, of course, requiring proportionately larger computing time). The process of line finding consists of determining edge magnitude and direction by convolution of an image with a number of edge masks, of thinning and thresholding of these edge magnitudes, of the linking of the edge elements based on proximity and orientation, and finally of approximation of the linked elements by piecewise linear segments.

Edge Detection

Edge detection is done by convolving a given image with masks corresponding to ideal step edges in a selected number of directions. The magnitude of the convolved output and the direction of the mask giving the highest output at each pixel are recorded as edge data. (The edge data are two files, one containing the magnitude and the other, a coded direction). We have found 5×5 masks in six directions as shown in Fig. 1 to be suitable for most images of interest. The choice of mask sizes needs to be investigated further. In general, the small masks are more sensitive to noise whereas the larger masks cannot resolve fine detail and may have difficulties if the texture elements are of similar size. We have chosen not to use the techniques of adaptive mask size selection by comparing the outputs of a large number of masks of varying size as suggested by Rosenfeld and Thurston [1] and by Marr [2], due to unacceptable computational requirements for large images. The criteria for choosing from among the many sizes are also unclear in

-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100
-100	-100	0	100	100

(a) 0°

-100	32	100	100	100
-100	-78	92	100	100
-100	-100	0	100	100
-100	-100	-92	78	100
-100	-100	-100	-32	100

(b) 30°

100	100	100	100	100
-32	78	100	100	100
-100	-92	0	92	100
-100	-100	-100	-78	32
-100	-100	-100	-100	-100

(c) 60°

100	100	100	100	100
100	100	100	100	100
0	0	0	0	0
-100	-100	-100	-100	-100
-100	-100	-100	-100	-100

(d) 90°

100	100	100	100	100
100	100	100	78	-32
100	92	0	-92	-100
32	-78	-100	-100	-100
-100	-100	-100	-100	-100

(e) 120°

100	100	100	32	-100
100	100	92	-78	-100
100	100	0	-100	-100
100	78	-92	-100	-100
100	32	-100	-100	-100

(f) 150°

Fig. 1. Edge Masks in 6 directions

presence of texture. However, use of more than one mask size may be necessary for certain applications.

Thinning and Thresholding

The presence of an edge at a pixel is decided by comparing the edge data with some of the 8 neighboring pixels. An edge element is said to be present at a pixel if:

1. the output edge magnitude at the pixel is larger than the edge magnitudes of its two neighbours in a direction normal to the direction of this edge. (The normal to a 30 degree edge is approximated by the diagonals on a 3×3 grid);
2. the edge directions of the two neighboring pixels are within one unit (30 degrees) of that of the central pixel and
3. the edge magnitude of the central pixel exceeds a fixed threshold.

Further, if the conditions 1 and 2 above are satisfied, the two neighboring pixels are disqualified from being candidates for edges. This algorithm produces results independent of the order in which the pixels are examined.

A more judicious decision could be based on examining the shape of the profile of convolution output, e.g., an ideal step edge should produce a triangle-shaped output. Such techniques have been used by Herskovitts and Binford [3] and by Marr [2]. Our experiments with requiring the neighboring pixels to have edge magnitudes that are at least a certain fraction of the central pixel magnitude resulted in poor performance perhaps due to variations caused by fine

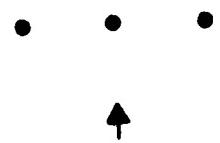
texture in the test images. More complex decision strategies hold promise for improved performance.

Linking

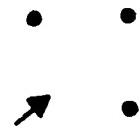
A boundary in a digital plane is a collection of points where each point is connected to two of its 8-neighbors. (Except for edge points and where "forks" exist). One approach to connect up such points, therefore, is to determine the two neighbors for each edge point. The two neighbors can be further distinguished as a predecessor and a successor. The boundary is then a threading through these edge points using this information.

The primary aspect of the linking process is the determination of a predecessor and a successor, if any, at each edge point. We produce two matrices - p and s - of the same physical dimensions as the image. (We have stored them as p and s files on the disk). Our criteria for connecting two edge points is that they be neighbors, in the 8-neighbor sense, and that they have edge directions differing by not more than a certain value, currently set at 30 degrees for masks described previously. Due to the nature of thinning, only three locations are potential candidates for predecessor or successor elements as shown in Figs. 2(a) and (b) for edges of 0 and 30 degree directions respectively. The determination of successor (predecessor) pixels is elaborate due to the several cases that are possible at each pixel:

1. Only one element is an acceptable successor.
In this case the successor (predecessor) is recorded in the s(p) file as an integer between 0 and 7 corresponding to its location.
2. Two candidates are acceptable successors. If



(a) 0° edge



(b) 30° edge

Fig. 2 Possible successor locations for two edges

they are not 4-neighbors, a fork is present as shown in Fig. 3(a). If they are 4-neighbors, a fork exists only if their directions differ by more than 2 units (60 degrees), as in Fig. 3(b). Otherwise no fork exists and the nearer of the two (using Euclidean distance), forms the successor (predecessor), as shown in Fig. 3(c).

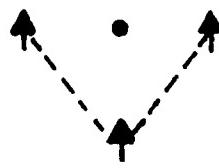
These rules are for smooth continuation of lines and were derived by complete enumeration of such configurations.

In case of a fork the stronger of the two candidates in edge magnitude forms the main stream. The fact that a fork exists is noted in the s(p) file. This information is sufficient to trace both streams of a fork by examining the p and s files simultaneously.

3. Three candidates are acceptable successors.

Fig. 4 shows all possible such configurations for a vertical edge (no three successor configurations occur for 30 degree edges). In these cases, a fork exists. The main stream is formed by the nearer of the two edges having the same direction, and the other candidate with different direction forms the other branch.

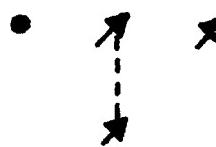
Note that this representation of the linked edge elements is in contrast to explicit lists of elements forming a connected segment. For large images, not entirely resident in core, it is more convenient to form predecessor and successor matrices as the processing requires only a sequential scan of the image file. Further, certain proximity computations can be more easily performed using the predecessor and successor files.



(a) Non-neighboring successors

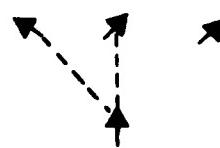


(b) Successors directions differ by 60°

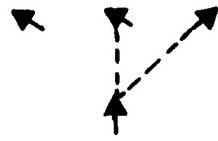


(c) Successors of same direction

Fig. 3 Three instances of two successors



(a)



(b)

Fig. 4 All instances of three successors

We now describe briefly how we can make use of the p and s matrices to produce a one-time traversing of all the curves in the picture. Such a traversing is necessary both to obtain a display on a suitable device and in fitting linear segments to the curves as described later. The general scheme is a TV raster scan which looks for the condition for starting a traversal:

```
var rscan: 1..noofrows;
      cscan: 1..noofcolumns;
for rscan := 1 step 1 until noofrows do
begin
  for cscan := 1 step 1 until noofcolumns do
begin
  if start(rscan,cscan) then
repeat
  visit this pixel;
  compute next pixel;
  until cannot proceed;
end;
end;
```

The above algorithm is applied to the p and s files in three passes, with a different predicate "start" to decide if traversing should start at a pixel. In the first case, a traversing starts when a pixel does not have a predecessor but has a successor. The second pass examines if the predecessor was a fork point, and thus picks up the secondary branches. The final pass starts traversing at those pixels that have not been "visited" previously and picks out circular segments. Information about previous visits is stored in a temporary binary file. During any pass, we "cannot proceed" if we come to a pixel that has already been visited.

Fitting Piecewise Linear Segments

If we are looking for straight edges in the picture, we need to fit piecewise linear segments to the (digital) curves that we obtain after linking, as described above. We have used a version of the iterative end-point fits algorithm of Duda and Hart [4]. A point on a digital curve is a corner if it is the most-removed from the endpoints. The first corner in a curve thus produces two segments both of which can contain more corners and so a recursive application of the same procedure is appropriate:

```
type point = record
    r: rowcoordinate;
    c: columncoordinate;
  end;
procedure cornersinwindow(p1, p2 : point);
var p: point;
begin
  p := p1;
  repeat
    p := next(p);
    if p is a corner then
      begin
        mark p as a corner;
        cornersinwindow(p1,p);
        cornersinwindow(p,p2);
      end;
    until p = p2;
end;
```

A straightforward application of such a recursive procedure can be inefficient. On an average, it takes $O(n^2)$ time to process a curve which is n points long. Hence, a variation which embodies the above mentioned qualities but is superior is employed. Instead of considering the entire

curve and then applying the above procedure we apply it on a smaller portion of the entire curve, say m points long. For the next part of processing, the curve begins at the farthest corner found, and ends m points later and so on, until the end of the original curve is reached. To avoid the possibility of the algorithm missing some corners because the end point of an m -long portion was at or around a genuine corner, we consider $2m$ -long chains in case no corners are found, then chains $3m$ -long...and so on, until either we find a corner or come to the end. A typical value for m is 32 elements. We believe this algorithm to substantially faster on the average, but have not yet performed a detailed analysis or comparison.

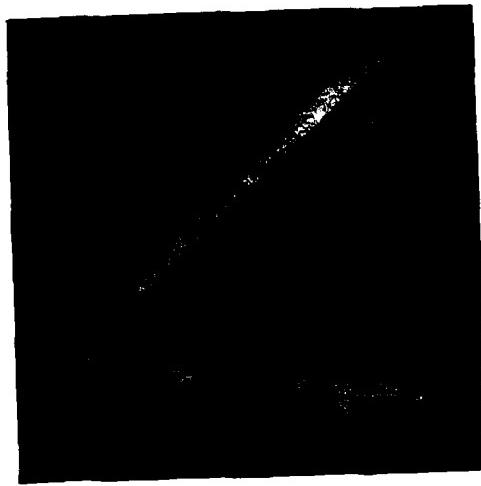
On output a segment is described by a unique id, its predecessor or successor segments along the flow of the curve, coordinates of the end points, length and direction.

Results

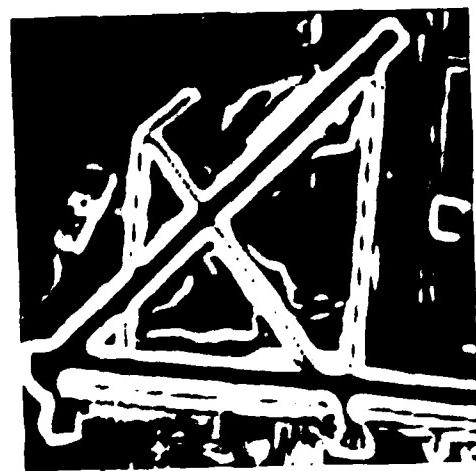
Results of processing an airport image at various stages of processing are shown in Figure 5. The computation times for various stages of processing are as follows (for a 128 x 128 image, on a PDP-10, KL-10 processor):

Convolution with edge masks	17 secs.
Thinning and Thresholding	2.3 secs.
linking (p and s files)	2.2 secs.
Segment tracing and linear approximations (maximum error-2 pixels)	4.8 secs.

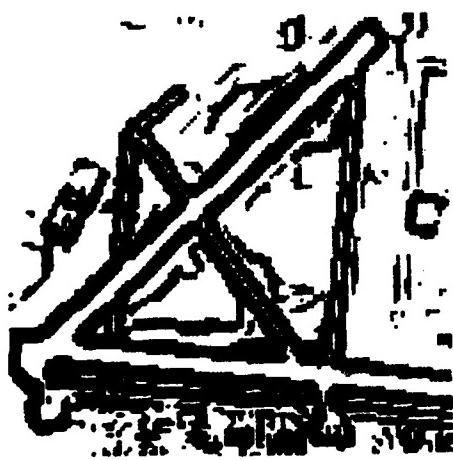
All computation times, except for linear segment fitting, scale linearly with the number of points to be



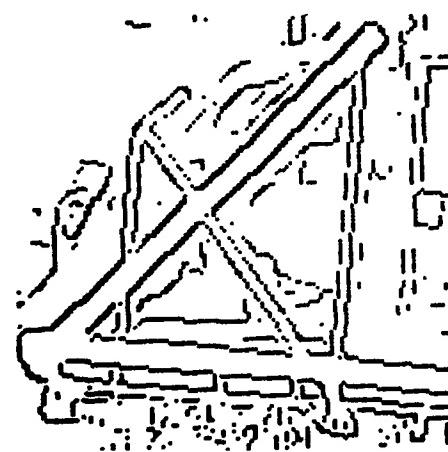
(a) Digital image



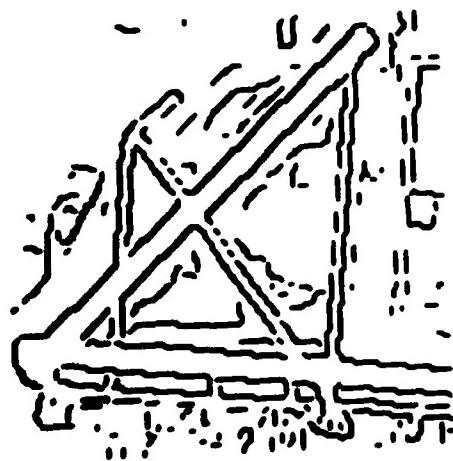
(b) Edge magnitude



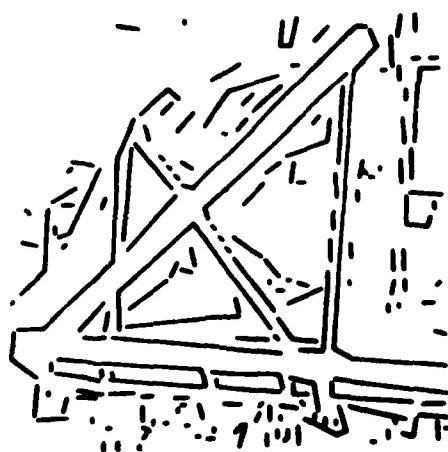
(c) Thresholded edges
(not thinned)



(d) Thinned and Thresholded
output



(e) Linked segments



(f) Linear segment
approximation

Fig. 5 An airport image and various stages of processing

processed. Also, except for the linear segment approximation, the storage requirements are limited to only a few lines of an image at a time.

References

1. A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," IEEE Transactions on Computers, Vol. 20, May 1971, pp. 562-569.
2. D. Marr, "Early Processing of Visual Information," MIT AI Memo No. 340, Dec. 1975.
3. A. Herskovitts, "On Boundary Detection," MIT AI Memo 183, 1970.
4. R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, Wiley, 1973.

2.9 Descriptions of Linear Segment Objects

K. Ramesh Babu and Ramakant Nevatia

An earlier section describes extraction of linear line segments from an image. These linear segments may suffice for isolating and recognizing certain useful objects, even in complex, natural imagery. We have chosen the problems of airport and road detection as test examples.

An airport can be modeled essentially by a set of its runways, taxiways and their relative dispositions. These runways and taxiways, and long segments of roads are bounded by long, parallel and piecewise linear segments of boundaries, and can be conveniently represented as "2-D generalized

"cones", with axes along the elongated direction and cross-sections along the width (see [1,2]).

In this section, we describe early attempts at computing generalized cone descriptions of elongated objects from the extracted boundary segments. Previous work on such descriptions has assumed availability of complete and perfect boundary information [1,3]. As objects of interest are thin and elongated we simply look for nearby parallel line segments of opposite directions, called antiparallel lines or apars, as potentially bounding a generalized cone of interest. Further selection of these cones may be based on their interrelationships and alternative descriptions may have to be carried through the matching process. In this section, we concentrate on the process of finding antiparallel line pairs.

Finding Antiparallel Pairs

The first thing that we do before pairing is to sort the segments by their orientation. This sorting collects together segments that are potential matches to a given segment and hence avoid looking through the entire list in finding a match for the given segment. However, due to errors in the orientation of segments, sorting based on exact angles is unnecessary, and the segments with the same angles correct to the nearest integer are grouped together.

In finding a pair of segments of antiparallel orientation we look for those whose angles are $(180+\alpha)^\circ$ apart, where α is a tolerance factor. Further, we require that the segments overlap and that they be within a certain distance of each other. These antiparallel pairs are then described as 2-dimensional generalized cones, with an axis and a width and an additional attribute of relative

brightness. A unique identifier is associated with each apar.

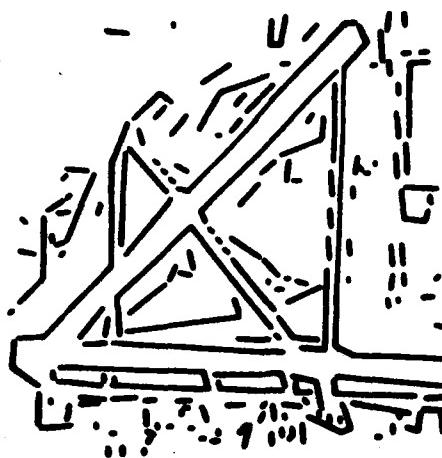
An output of the process of finding apars is given in Fig. 1(b) in which each apar is depicted as a 2-dimensional generalized cone (a rectangle). Fig. 1(a) is the input of line segments.

Selection among Antiparallels

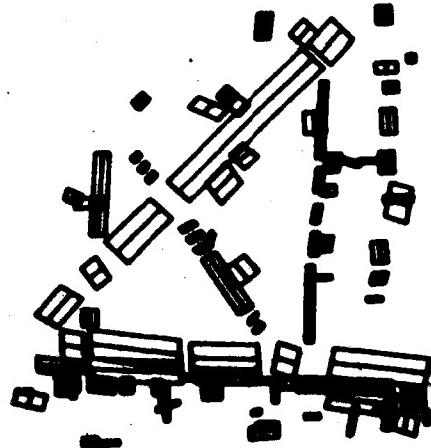
Not all antiparallels necessarily correspond to objects. In Fig. 2(a), for example, we have bc, ac, cd, and ad as apars. We retain only bc as a useful apar on the assumption that objects do not have an edge along their length.

If information about the relative brightness of the objects is known a priori, we can effect further reduction in the number of apars to be used for later, higher-level, processing. As an example, in Fig. 2(b), ef, gh and eh are brighter than the background according to our convention while fg is darker. In fact, fg lies in the region of eh. Now, if we know that our objects are brighter than the regions surrounding them, fg as an apar representing an object is ruled out. Further eh can be distinguished from ef and gh as being bounded by segments closest to each other. This distinction may be useful for locating runways, taxiways and roads.

In general, proper choice of apars is like resolving the figure-ground relationships. The difficulty is compounded if some of the line segments are missing due to inadequacies of lower level algorithms.



(a) line segments



(b) antiparallels (displayed as rectangles)

Fig. 1. Finding Antiparallel Segments



(a)

(b)

Fig. 2. Some antiparallels, illustrating difficulties of object isolation

At this point in the chain of processing the image, we have a list of apars which potentially correspond to portions of runways and taxiways. We now group the apars which are collinear, within certain angular and spatial tolerances, and where the gap between axes is less than a certain value. Such gaps are caused by intersecting runways and by errors of previous processing.

At the time of writing, the implementation of finding collinears is in progress.

References

1. R. Nevatia and T.O. Binford, "Description and Recognition of Curved Objects". Artificial Intelligence, Vol. 8, No. 1, February 1977, pp. 77-98.
2. R.A. Brooks, R. Greiner and T.O. binford, "A Model Based Vision System", Proceedings of ARPA Image Understanding Workshop, May 1978, pp. 36-41.
3. D. Marr, "Analysis of Occluding Contour", MIT AI Memo 372, October 1976.

3. Image Processing Projects

A variety of image processing projects are reported herein. They fall into three general areas of computational procedures, restoration methodologies, and inverse SAR imaging. A presentation is made on the computation of the condition number of a matrix to predict the degree of ill-conditioning and subsequent potential degrees of freedom in such a process. Such computations become extremely useful for large matrix processes as found in most imaging applications. In the generation of computer hologram interpolations, a special computational savings is developed to avoid the inefficiencies of zero padding traditionally used in most Fourier image filtering techniques.

In the arena of image restoration two techniques are reported upon. Results from the method of blind a posteriori restoration are presented in pictorial form. A new method of Poisson MAP restoration is also developed and analysis presented in which improved sensor models for imaging result.

Finally, two papers on inverse synthetic aperture radar imaging are presented. One is formative in its presentation and proposes to image shadowed regions via RATSCAT turntable data. The second represents processing results from an inflight aircraft in both a straight flight and a turn set of geometries. Resulting imagery is presented.

3.1 Condition Number Computation Of A Discrete Deconvolution Operator

William K. Pratt

Introduction

In many digital signal processing operations, it is necessary to deconvolve an observed signal that has been previously convolved with a known impulse response. The more general task is to estimate the original signal by linear or nonlinear processing of the observation. The accuracy of the deconvolution or estimation process is often dependent upon the "condition" of the deconvolution operator in the sense that small perturbations in measuring or modelling the impulse response in conjunction with observation uncertainty may lead to very large perturbations of the output signal. Signal processing accuracy is often expressed in terms of the condition number of the deconvolution operator. Conventional formulations of the condition number are of theoretical interest, but, in practice, are often useless because of the great amount of computation required for their calculation. In fact, the condition number computation itself is subject to inaccuracy as a result of ill-conditioning!

This paper introduces an easily computed formulation of the condition number of a deconvolution operator. Examples are given for one-dimensional and two-dimensional convolution.

Convolution operators

There are a variety of convolution operators used in digital signal processing [1,2]. The most basic is the

finite length operator defined by

$$q(m) = \sum_{n=1}^m f(n)h(m-n+1) + \epsilon(m) \quad (1)$$

where f is an N element input sequence, q is an M element output sequence, and h is an L element impulse response sequence with $M = N+L-1$, and $\epsilon(m)$ represents noise or measurement uncertainty at the m -th observation. This operation can also be cast in the vector space form

$$\underline{q} = \underline{D} \underline{f} + \underline{\epsilon} \quad (2)$$

where \underline{q} and $\underline{\epsilon}$ are $M \times 1$ column vectors, \underline{f} is an $N \times 1$ vector, and \underline{D} is an $M \times N$ matrix with general element

$$D(m,n) = h(m-n+1) \quad (3)$$

A two-dimensional formulation of the convolution operator is easily obtained by column-scanning the input and output arrays \underline{F} and \underline{Q} to produce the corresponding linked vectors \underline{f} and \underline{q} for direct entry into eq.(2). The two-dimensional convolution matrix \underline{D} is a partitioned matrix defined as

$$D_{m_2, n_2}(m_1, n_1) = H(m_1 - n_1 + 1, m_2 - n_2 + 1) \quad (4)$$

where \underline{H} is the $L \times L$ impulse response matrix.

Deconvolution

Deconvolution is the inversion process in which an estimate

$$\hat{\underline{f}} = \underline{W} \underline{q} \quad (5)$$

is formed by linear processing of the observation of eq.(2) with an $N \times M$ matrix \underline{W} . In general, $\hat{\underline{f}} \neq \underline{f}$ because \underline{W} is not a left inverse of \underline{D} and because of the noise residual $\underline{W} \underline{\epsilon}$. A minimum norm, least squares error estimate can be obtained

with a generalized inverse operator [3-5] $\underline{W} = \underline{D}^-$ satisfying the relations

$$\underline{D} \underline{D}^- \underline{D} = \underline{D} \quad (6a)$$

$$\underline{D}^- \underline{D} \underline{D}^- = \underline{D}^- \quad (6b)$$

$$\underline{D} \underline{D}^- = (\underline{D} \underline{D}^-)^T \quad (6c)$$

$$\underline{D}^- \underline{D} = (\underline{D}^- \underline{D})^T \quad (6d)$$

The accuracy of the deconvolution estimate can be bounded in terms of the noise perturbation $\underline{\epsilon}$. It can be shown [6] that

$$\frac{\|\underline{\Delta f}\|}{\|\underline{f}\|} \leq \|\underline{D}\| \cdot \|\underline{D}^-\| \frac{\|\underline{\epsilon}\|}{\|\underline{g}\|} \quad (7)$$

where $\|\cdot\|$ represents a vector or matrix norm and $\underline{\Delta f}$ denotes the perturbation of the correct solution. Equation (7) also suffices as an error bound for a perturbation $\underline{\Delta D}$ in the convolution operator by replacing $\underline{\epsilon}$ by $\underline{\Delta D} \underline{f}$. The term

$$C(\underline{D}) = \|\underline{D}^-\| \cdot \|\underline{D}\| \quad (8)$$

in eq.(7) is called the condition number of the operator. Basically, it represents an undesired amplification of system errors.

There are several definitions of the condition number in common usage that are based on different specifications of the matrix norm [7]. One of the most common is the least squares definition for which the matrix norm is given by

$$\|\underline{D}\| = \left[\sum_{m=1}^M \sum_{n=1}^N |D(m,n)|^2 \right]^{\frac{1}{2}} = \left[\text{tr}\{\underline{D} \underline{D}^{*T}\} \right]^{\frac{1}{2}} = \left[\text{tr}\{\underline{D}^{*T} \underline{D}\} \right]^{\frac{1}{2}} \quad (9)$$

Another useful formulation of this norm can be obtained by a singular value decomposition of the matrix

$$\underline{D} = \underline{U} \underline{S} \underline{V}^{*T} \quad (10)$$

where \underline{U} and \underline{V} are unitary matrices and \underline{S} is a diagonal matrix whose elements $S(1) \geq S(2) \geq \dots \geq S(R) \geq 0$ are singular values of \underline{D} . Since the trace operation of eq.(9) is invariant to the unitary transformation of eq.(10), the matrix norm can be rewritten as

$$\|\underline{D}\| = \sum_{n=1}^R |S(n)|^2 \quad (11)$$

where R is the rank of \underline{D} . Then, since

$$\underline{D}^{-1} = \underline{V} \underline{S}^{-1} \underline{U}^{*T} \quad (12)$$

the matrix norm of \underline{D}^{-1} can be immediately expressed as

$$\|\underline{D}^{-1}\| = \sum_{n=1}^R |S(n)|^{-2} \quad (13)$$

This leads to the conventional formulation of the condition number

$$C(\underline{D}) = \left[\sum_{n=1}^R |S(n)|^2 \sum_{n=1}^R |S(n)|^{-2} \right]^{\frac{1}{2}} \quad (14)$$

This definition of condition number deliberately avoids the occurrence of an infinite value caused by a zero valued singular value. The rationale is that the deconvolution processor should be designed so that it avoids direct inversion when \underline{D} contains zero-valued singular values.

Deconvolution operator condition number

The condition number of eq.(8) can be computed in a brute force manner by first generating the generalized inverse \underline{D}^{-1} from \underline{D} and then computing the matrix norm from eq.(9). But, if \underline{D} is large, generation of \underline{D}^{-1} directly can be extremely time-consuming on a general purpose computer.

(Note that the actual deconvolution operation can often be accomplished by indirect means that do not require explicit generation of \underline{D}^{-1} .) Alternatively, the condition number can be computed in terms of the non-zero singular values of \underline{D} . But, singular value generation by computational methods is also impractical for large size operators. This has been the motivation to develop a more easily computed formulation of the condition number of deconvolution operators. The concept is quite simple: the singular value expansion of eq.(10) is replaced by an analytic Fourier transform expansion [8].

The Fourier domain representations of a convolution matrix \underline{D} and its generalized inverse \underline{D}^{-1} are defined as [1]

$$\underline{\mathcal{D}} = \underline{A}_M \underline{D} \underline{A}_N^{-1} \quad (15a)$$

$$\underline{\mathcal{D}}^{-1} = \underline{A}_N \underline{D}^{-1} \underline{A}_M \quad (15b)$$

where, for a one-dimensional operator, \underline{A} is an $N \times N$ matrix with a general term $A(u,j) = \gamma_N^{(j-1)(u-1)}$ with $\gamma_N = \exp\{-2\pi i/N\}$ for $1 < j, u < N$. For a two-dimensional operator, \underline{A} is replaced by $\underline{A} \otimes \underline{A}$ where \otimes denotes a left-direct product. Matrix $\underline{\mathcal{D}}$ can be represented analytically as [1]

$$\underline{\mathcal{D}} = \underline{\chi}_M \underline{\varphi} \quad (16)$$

where $\underline{\chi}_M$ is an $M \times M$ diagonal matrix whose diagonal terms are elements of the transfer function

$$h(u) = \frac{1}{\sqrt{M}} \sum_{j=1}^L h(j) \exp\left\{\frac{-2\pi i}{M}(u-1)(j-1)\right\} \quad (17)$$

defined as the one-dimensional Fourier transform of the impulse response for $1 \leq u \leq M$. For two-dimensional

convolution, the terms of $\underline{\mathcal{X}}_M$ are the column-scanned elements of the two-dimensional Fourier transform $\underline{\mathcal{X}}$ of the impulse response H . The interpolation matrix is given by [1]

$$\underline{\varrho}(u,v) = \frac{1}{\sqrt{N}} \frac{1-\underline{\mathcal{X}}_M^{-(u-1)(L-1)}}{1-\underline{\mathcal{X}}_M^{(u-1)}\underline{\mathcal{X}}_N^{-(v-1)}} \quad (18)$$

For two-dimensional convolution, $\underline{\varrho}$ is replaced by $\underline{\varrho} \otimes \underline{\varrho}$. It is important that the length M of the convolution output and the length L of the impulse response be properly chosen to avoid zero-valued terms in \underline{h} or $\underline{\mathcal{X}}$ caused by truncation of the impulse response. This can be accomplished by setting L to an odd integer and zero padding the input data f or F , if necessary, such that $M = 2^m$ where m is an integer.

The next step is to compute the norms of \underline{D} and its generalized inverse \underline{D}^- in terms of the Fourier domain representations $\underline{\varrho}$ and $\underline{\varrho}^-$. Since the Fourier transform is unitary, the norms become

$$\|\underline{D}\|^2 = \text{tr}\{\underline{\mathcal{X}}_M \underline{\varrho} \underline{\varrho}^{*T} \underline{\mathcal{X}}_M^*\} = \text{tr}\{\underline{\varrho}^{*T} \underline{\mathcal{X}}_M^* \underline{\mathcal{X}}_M \underline{\varrho}\} \quad (19a)$$

$$\|\underline{D}^{-}\|^2 = \text{tr}\{(\underline{\mathcal{X}}_M \underline{\varrho} \underline{\varrho}^{*T} \underline{\mathcal{X}}_M^*)^{-}\} = \text{tr}\{\underline{\varrho}^{*T} \underline{\mathcal{X}}_M^* \underline{\mathcal{X}}_M \underline{\varrho}\}^{-} \quad (19b)$$

The matrix norms can be expressed in terms of the interpolation matrix product $\underline{\vartheta} = \underline{\varrho} \underline{\varrho}^{*T}$. It can be shown that for one-dimensional convolution

$$\underline{\vartheta}(u,w) = \exp\left\{-\frac{i\pi}{M}(N-1)(u-w)\right\} \frac{\sin\left[\frac{\pi N}{M}(u-w)\right]}{\sin\left[\frac{\pi}{M}(u-w)\right]} \quad (20)$$

For two-dimensional convolution, $\underline{\vartheta}$ is replaced by its self-direct product.

From these expressions the matrix norms of \underline{D} is immediately found to be

$$\|\underline{D}\|^2 = N \sum_{m=1}^M |\underline{\chi}_M(m,m)|^2 \quad (21)$$

But, from the discrete form of Parseval's theorem, the matrix norm can be trivially evaluated from the convolution impulse response. Thus, for one-dimensional convolution

$$\|\underline{D}\|^2 = N \sum_{j=1}^L |h(j)|^2 \quad (22a)$$

and for two-dimensional convolution

$$\|\underline{D}\|^2 = N^2 \sum_{j=1}^L \sum_{k=1}^L |H(j,k)|^2 \quad (22b)$$

Computation of $\|\underline{D}^{-1}\|^2$ is not so simple, unfortunately, because of the generalized inverse operations of eq.(19b). No exact closed form expression has been found for this equation. But, a closed form approximation can be obtained by replacement of the generalized inverse norm $\|\underline{D}^{-1}\|^2$ by the less restrictive conditional inverse norm $\|\underline{D}^{\#}\|^2$. It is easily verified that

$$(\underline{\chi}_M \ \underline{\epsilon} \ \underline{\chi}_M^{*T})^{\#} = (\underline{\chi}_M^{*})^{-1} \ \underline{\epsilon}^{-1} (\underline{\chi}_M)^{-1} \quad (23)$$

is a conditional inverse satisfying conditions A and B of eq.(6), but not conditions C and D. Furthermore, it is observed that $\underline{\epsilon}$ is an imdepotent matrix to within a scale factor. Hence, the generalized inverse can be written immediately as

$$\underline{\mathcal{L}}^{-} = \frac{1}{M^2} \underline{\mathcal{L}} \quad (24)$$

As a consequence, the conditional inverse norm for one-dimensional convolution can be directly computed according to the formula

$$\|\underline{\mathcal{D}}^{\#}\|^2 = \sum_{u=0}^{M-1} \frac{N}{M^2} \left[|h(u)|^2 \right]^{-} \quad (25a)$$

and for two-dimensional convolution

$$\|\underline{\mathcal{D}}^{\#}\|^2 = \sum_{u=0}^{M-1} \sum_{v=0}^{M-1} \frac{N^2}{M^4} \left[|\mathcal{H}(u,v)|^2 \right]^{-} \quad (25b)$$

The approximate condition number formula then becomes

$$C(\underline{\mathcal{D}}) = \|\underline{\mathcal{D}}\| \cdot \|\underline{\mathcal{D}}^{\#}\| \quad (26)$$

where the norm of the convolution operator is given by eq.(22) and the norm of the conditional inverse of the operators by eq.(26).

Evaluation

The condition number formulation given by eq.(26) is an approximation to the standard definition of eq.(8) because of the use of the conditional rather than the generalized inverse. Consideration is now given to the effects of the approximation.

The usefulness of any matrix condition number is its ability to predict numerical errors that may occur in

computation. Condition numbers are rarely used as precise metrics of the ill-conditioning of a matrix, but rather as a rough indication of potential ill-conditioning problems. Accordingly, the procedure taken to evaluate the conditional inverse formulation has been to calculate the generalized inverse and conditional inverse forms for several impulse response operators, and then to relate the condition number values to the observed computational error in some deconvolution experiments.

The simplest type of convolution operator is the moving window average for which the one-dimensional impulse response is $h(j) = 1/L$ for $1 \leq j \leq L$, and in two dimensions, $H(j,k) = 1/L^2$ for $1 \leq j,k \leq L$. The corresponding transfer functions are

$$h(u) = \frac{1}{\sqrt{J}} \exp\left\{-\frac{i\pi(L-1)}{J}(u-1)\right\} \cdot \frac{\sin\left[\frac{\pi L}{J}(u-1)\right]}{\sin\left[\frac{\pi(u-1)}{J}\right]} \quad (27a)$$

$$h(u,v) = h(u) h(v) \quad (27b)$$

for $1 \leq u, v \leq M$. The separable Gaussian-shaped impulse response is another simple, widely used convolution operator. Even in these simple cases, no simple closed forms have been found for the generalized inverse condition number; evaluation must be performed either directly by eq.(8) or indirectly by the singular value decomposition formula of eq.(14). The latter approach has been taken to evaluate the exact condition number for one-dimensional convolution with an impulse response length of $L = 5$ for various lengths of the input sequence. Results are plotted in figure 1 for the generalized and conditional inverse formulations. The agreement is seen to be reasonably good.

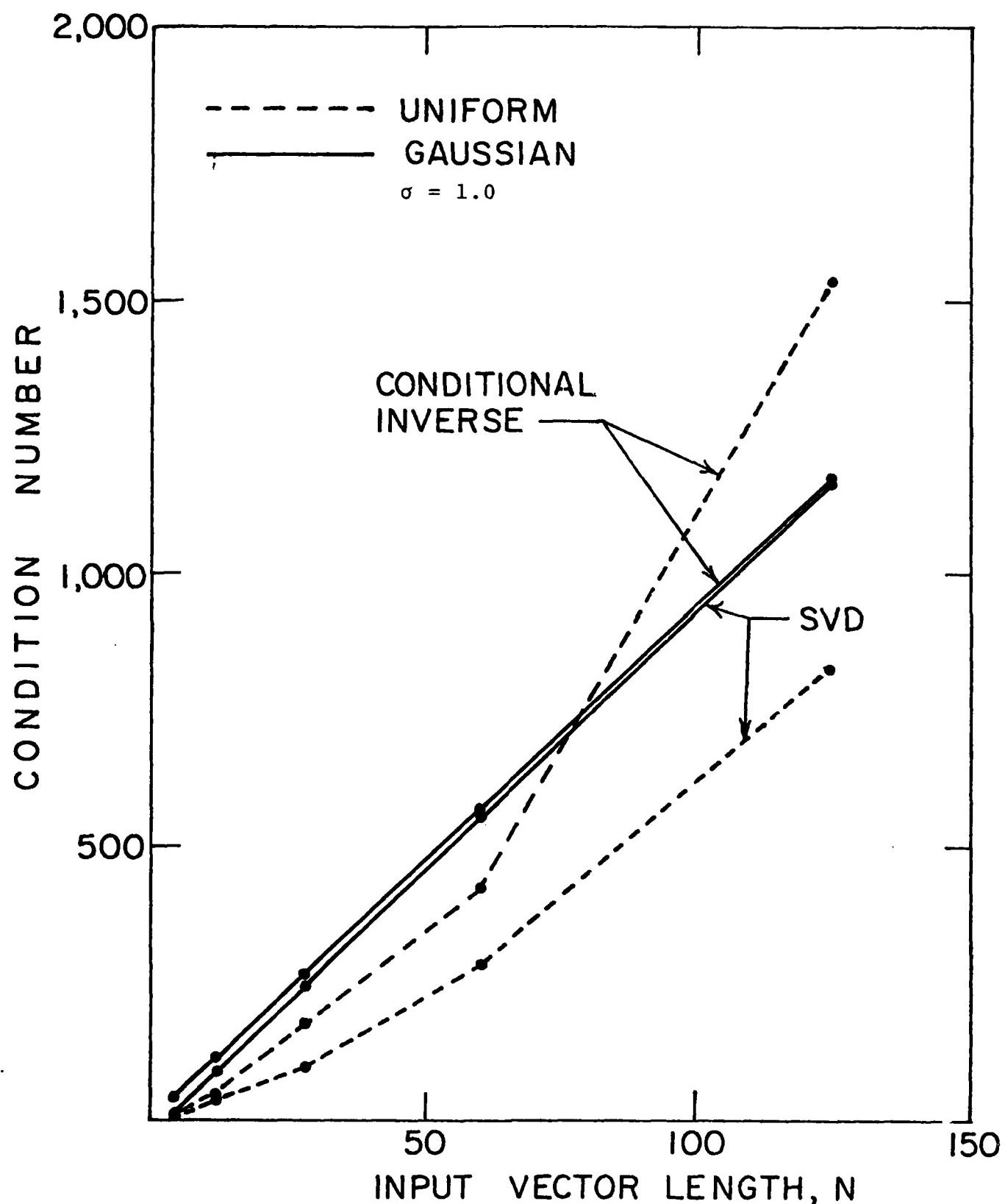


Figure 1. Comparison of generalized and conditional inverse condition numbers for one-dimensional uniform and Gaussian-shaped impulse response, L=5.

Figure 2 presents examples of the conditional inverse condition number of several two-dimensional deconvolution operators, as a function of the size of the impulse response for an output image of 64 x 64 samples. The impulse responses tested are: L x L square and square annular apertures and circular and circular annular apertures of diameter L. The annular apertures are of one pixel width.

Conclusions

A new formulation of the condition number of a discrete convolution operator has been introduced. This formulation, based on the conditional inverse of a deconvolution operator, is extremely simple to compute. Computation requires a weighted summation of a single K-dimensional Fourier transform of a K-dimensional impulse response. Examples presented show that the conditional inverse condition number is in reasonable agreement with the conventional formulation based on the generalized inverse of a deconvolution operator.

Acknowledgement

This work was performed while the author was on Sabbatical leave as a Visiting Professor at the Institut d'Optique, Universite de Paris-Sud, Paris, France. The stimulating environment supplied by Professor S. Lowenthal of the Institut d'Optique is well appreciated. Computer programming and Experiments were performed by Mr. Sang Uk Lee of USC.

References

1. W.K. Pratt, "Vector Space Formulation of Two-Dimensional Signal Processing Operations," Journal

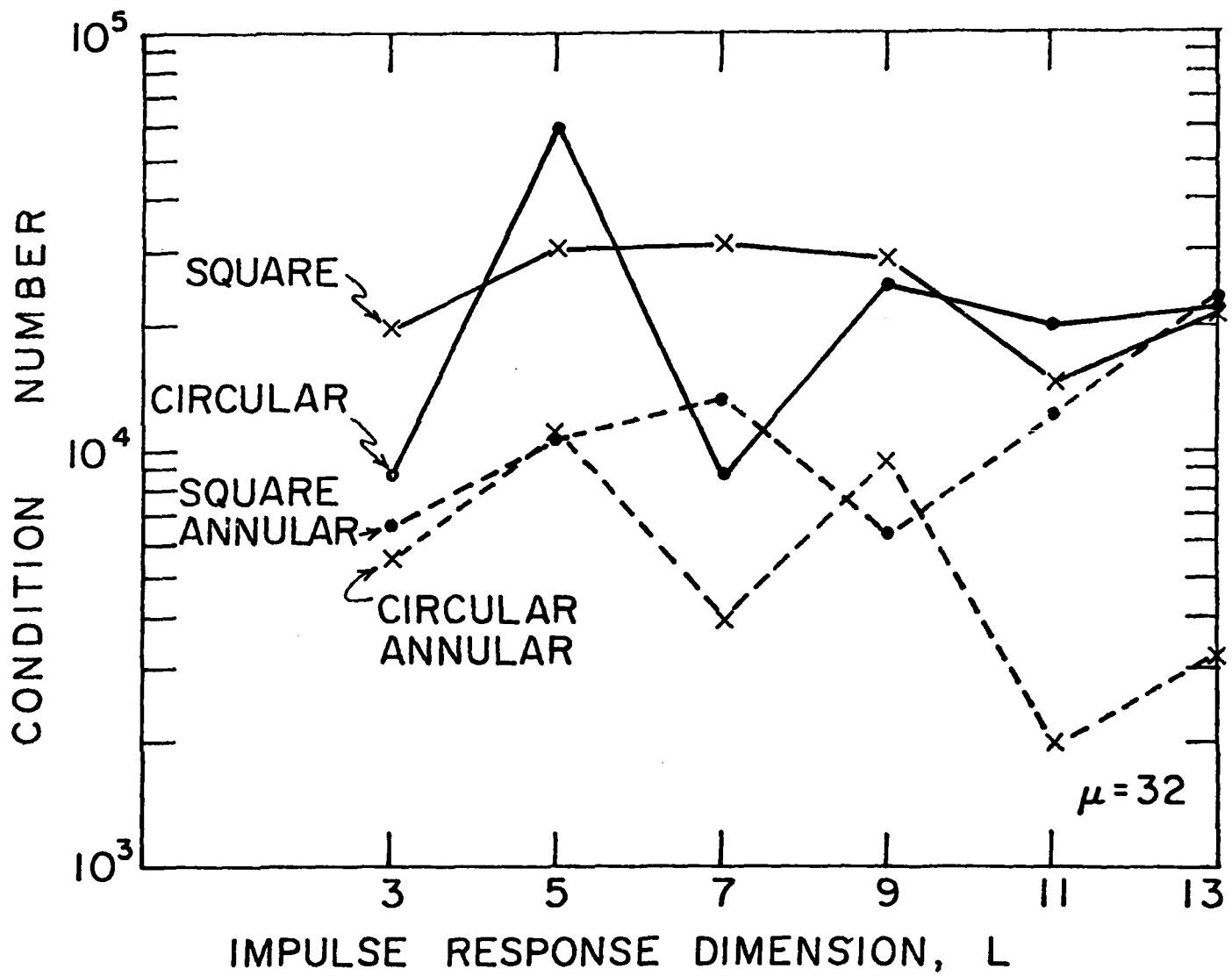


Figure 2. Conditional inverse condition numbers for several two-dimensional impulse responses.

Computer Graphics and Image Processing, Academic Press, New York, Vol. 4, No. 1, 1975, pp. 1-24.

2. W.K. Pratt, Digital Image Processing, Wiley-Interscience, New York, 1978.

3. A. Albert, Regression and the Moore-Penrose Pseudoinverse, Academic Press, New York, 1972.

4. C.R. Rao and S.K. Mitra, Generalized Inverse of Matrices and Its Applications, Wiley, New York, 1971.

5. F.A. Graybill, Introduction to Matrices with Applications in Statistics, Wadsworth, Belmont, Cal., 1969.

6. B.W. Rust and W.R. Burrus, Mathematical Programming and the Numerical Solution of Linear Equations, American Elsevier, New York, 1972.

7. D.K. Fadeev and V.N. Fadeeva, Computational Methods of Linear Algebra, Freeman, San Francisco, 1963.

8. W.K. Pratt, "Generalized Wiener Filtering Computation Techniques," IEEE Transactions on Computers, Vol. C-21, No. 7, July 1972, pp. 636-641.

3.2 Estimation of Image Signal with Poisson Noise - I

Chun-Moo Lo and Alexander A. Sawchuk

Introduction

The objective of this work is to develop an optimal filter (in the sense of maximum a posteriori probability (MAP)) for the output of an optical photo detector with photo electron shot noise. Since Hunt [6] introduced MAP estimation to image restoration, more researchers have used MAP estimation in image restoration. The MAP method can be generalized to linear or nonlinear image models and to noise models different from additive Gaussian noise. In addition, the MAP estimation can be a local adaptive processing method. The MAP filter contains a ML (maximum likelihood) term and an a priori nonstationary mean term based on the local nonstationary statistical properties of the images. Also the MAP filter can be extended to the case of space-variant degradations. Detailed results for additive Gaussian noise models have been given in [7] [8]. The most basic source of photo detector shot noise (poisson noise) lies in the photon fluctuations associated with the detection of the finite amount of light energy available to the imaging system. Thus photon fluctuations pose a fundamental limitation to the restoration of a degraded image. Of course, the statistical properties of poisson noise are quite different from additive noise or multiplicative noise. Hence conventional linear filtering can not be directly used. It is the purpose of this work to develop a MAP filter to process images with the Poisson noise model. This report describes the restoration non-blurred image signals with a Poisson noise model. Additional results for blurred images will be given later.

MAP Estimation for Non-blurred Image Signals with a Poisson Noise Model

Bayes' law leads immediately to the description of the a posteriori conditonal density

$$P(\underline{f}|\underline{d}) = P \frac{(\underline{d}|\underline{f}) P(\underline{f})}{P(\underline{d})} \quad (1)$$

where \underline{f} is original image which we want to estimate, and \underline{d} is the display intensity.

The use of the posterior density for estimation is well known. Minimum mean-square error (MMSE) estimates are the mean of the posterior density. Maximum a posteriori (MAP) estimates are the mode of the posterior density. Maximum likelihood (ML) estimates may be viewed as a special case of MAP estimate when a posterior density is equal to the a priori density [3]. However, MMSE is a nonlinear estimator. It needs to know the probability density of the observation data $p(\underline{d})$ which is usually impossible to get in practice. Usually linear minimum mean-square error (LMME) estimates are used. Goodman has worked on Poisson models with LMME estimate in [10]. Burke uses ML estimation on the Poisson model in [11]. The MAP estimate tries to find the value of \underline{f} which maximizes the posterior density $p(\underline{f}|\underline{d})$. It does not need $p(\underline{d})$ at all but does need $p(\underline{d}|\underline{f})$ and $p(\underline{f})$. To simplify problems of dimensionality, we first describe a single counter and later extend it to the whole array.

Single Counter

The model of a single counter is shown as fig. 1. According to the semi-classical theory of photo detection, the probability that g photon events occur given intensity f_i is

$$P(g_i|f_i) = \frac{(\lambda f_i)^{g_i} e^{-\lambda f_i}}{g_i!} \quad (2)$$

From (2) we can get

$$\bar{g}_i = \lambda f_i \quad \sigma_{g_i}^2 = \lambda f_i$$

$$\frac{\bar{g}_i}{f_i} = \lambda = \frac{\text{average # of count}}{\text{intensity unit}}$$

Here g_i is the photon counts, having no units, and we display an intensity $d_i = \alpha g_i$, where α is a scale factor. We usually choose $\alpha\lambda = 1$ in order to keep the mean value of the processed displayed signal the same as the observed noisy image. So

$$P(d_i | f_i) = \frac{1}{\alpha} P(d_i/\alpha | f_i) = \frac{(\lambda f_i)^{d_i/\alpha} e^{-\lambda f_i}}{d_i!} \quad (3a)$$

and we find that

$$\bar{d}_i = \alpha \bar{g}_i = \alpha \lambda f_i$$

$$\sigma_{d_i}^2 = \alpha^2 \sigma_{g_i}^2 = \alpha^2 \lambda f_i$$

so that

$$(S/N)_{rms} = \frac{d_i}{\sigma_{d_i}} = \frac{\alpha \lambda f_i}{\alpha (\lambda f_i)^{\frac{1}{2}}} = (\lambda f_i)^{\frac{1}{2}}$$

Thus, the $(S/N)_{rms}$ is proportional to the square root of the signal $(f_i^{\frac{1}{2}})$, and is signal dependent.

If f_i is a random variable as it is in an image then the mean of $(S/N)_{rms}$ given by $\overline{(S/N)}_{rms} = (\lambda \bar{f}_i)^{\frac{1}{2}}$ makes more sense. Thus Poisson noise is quite different from additive noise or multiplicative noise in its statistical properties as shown in Fig. 2. Here the upper left picture is the original image, the upper right picture is a Poisson noisy picture with $\overline{(S/N)}_{rms} = 6$ db. The lower left picture is a linear additive Gaussian noisy picture $\overline{(S/N)}_{rms} = 6$ db. Lower right is linear additive Gaussian noisy picture with

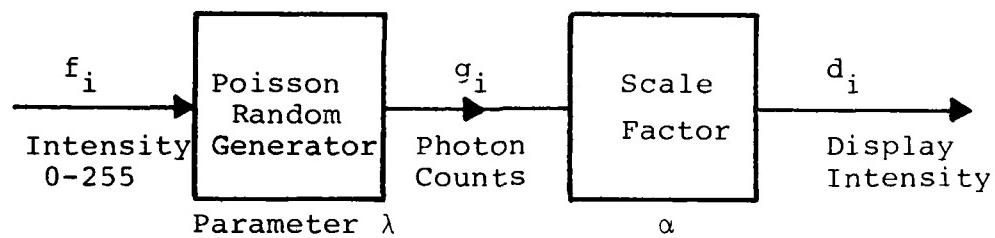


Fig. 1 Single Counter Model



Fig. 2 Poisson Noise and Additive Gaussian Noise

$(S/N)_{rms} = 0$ db. From these images it can be seen that Poisson noisy pictures are degraded more severely than linear additive noisy pictures, even though the $(S/N)_{rms}$ is 6 db lower than that of Poisson noisy picture.

Array Counters

An array counter model for non-blurred image signals with Poisson noise is shown as Fig. 3. For one single counter the conditional density is

$$P(g_i | f_i) = \frac{(\lambda f_i)^{g_i} e^{-\lambda f_i}}{g_i!} \quad (3b)$$

For array counters, we assume that all g_i are independent for given f (i.e. every Poisson generator is independent) and each g_i depends only on its corresponding f_i where

$$\underline{g} = [g_1, g_2, \dots, g_N]^T \text{ and } \underline{f} = [f_1, f_2, \dots, f_N]^T$$

Hence

$$P(\underline{g} | \underline{f}) = P(g_1 | f_1) P(g_2 | f_2) \dots P(g_N | f_N)$$

also each g_i depends only on its corresponding f_i , then

$$P(\underline{g} | \underline{f}) = \prod_i \frac{(\lambda f_i)^{g_i} e^{-\lambda f_i}}{g_i!} \quad (4)$$

now from before we know

$$\begin{aligned} P(\underline{d} | \underline{f}) &= P(d_1 | f_1) P(d_2 | f_2) \dots P(d_N | f_N) \\ &= P(d_1 | f_1) P(d_2 | f_2) \dots P(d_N | f_N) \end{aligned}$$

then

$$P(\underline{d} | \underline{f}) = \prod_i \frac{[(\lambda f_i)^{d_i/\alpha} e^{-\lambda f_i}]^{\alpha}}{\alpha(d_i/\alpha)!} \quad (5)$$

A Priori Density of Image $p(f)$

From [6], it is shown that a multivariate normal probability density with a non-stationary mean but with stationary covariance for $p(f)$ can still lead to useful results in processing real image signals, even signals which are non-Gaussian. Hence we assume

$$P(\underline{f}) = k_f \exp\left\{-\frac{1}{2}[(\underline{f}-\bar{\underline{f}})^T R_f^{-1}(\underline{f}-\bar{\underline{f}})]\right\} \quad (6)$$

where $\bar{\underline{f}}$ is non-stationary mean vector and R_f is covariance matrix.

MAP Estimate Equations

From (1), we maximize (1) with respect to \underline{f}

$$\underset{\underline{f}}{\text{Max}} P(\underline{f} | \underline{d}) = \frac{P(\underline{d} | \underline{f}) P(\underline{f})}{P(\underline{d})}$$

As is conventional, we work with the logarithm of the posterior density.

$$\underset{\underline{f}}{\text{Max}} [\ln P(\underline{f} | \underline{d})] = \ln P(\underline{d} | \underline{f}) + \ln P(\underline{f}) - \ln P(\underline{d})$$

Since the last term $\ln P(\underline{d})$ on the right does not depend on \underline{f} , we neglect it in maximization with respect to \underline{f} . Thus MAP estimate equation is given by

$$\frac{\partial \ln P(\underline{f} | \underline{d})}{\partial \underline{f}} = \frac{\partial \ln P(\underline{d} | \underline{f})}{\partial \underline{f}} + \frac{\partial \ln P(\underline{f})}{\partial \underline{f}} = \underline{o}^T \quad (7)$$

From (5), we have

$$\ln P(\underline{d} | \underline{f}) = \sum_i \ln \left[\frac{(\alpha f_i)^{d_i} / \alpha e^{-\lambda f_i}}{\alpha (\frac{d_i}{\alpha})!} \right]$$

$$= \sum_i \left\{ \frac{d_i}{\alpha} \ln(\lambda f_i) - \lambda f_i - \ln \alpha - \ln \left[\left(\frac{d_i}{\alpha} \right)! \right] \right\}$$

From (6), we have

$$\ln P(\underline{f}) = \ln k_f - \frac{1}{2} (\underline{f} - \bar{\underline{f}})^T R_f^{-1} (\underline{f} - \bar{\underline{f}})$$

Differentiating the above two equations individually with respect to \underline{f} we get

$$\frac{\partial \ln P(\underline{d} | \underline{f})}{\partial \underline{f}} = \left[\frac{d_1}{\alpha f_1} - \lambda, \frac{d_2}{\alpha f_2} - \lambda, \dots, \frac{d_N}{\alpha f_N} - \lambda \right] \quad (8)$$

and

$$\frac{\partial \ln P(\underline{f})}{\partial \underline{f}} = -\frac{1}{2} \cdot 2 (\underline{f} - \bar{\underline{f}})^T R_f^{-1} = -(\underline{f} - \bar{\underline{f}})^T R_f^{-1} \quad (9)$$

Substituting (8) and (9) into (7), we get

$$\left[\frac{d_1}{\alpha f_1} - \lambda, \frac{d_2}{\alpha f_2} - \lambda, \dots, \frac{d_N}{\alpha f_N} - \lambda \right] - (\underline{f} - \bar{\underline{f}})^T R_f^{-1} = \underline{o}^T$$

Taking the transpose on both sides of above equation and assuming R_f^{-1} is a symmetric matrix, (i.e. $R_f^{-1} = (R_f^{-1})^T$) then we get

$$\begin{bmatrix} \frac{d_1}{\alpha f_1} - \lambda \\ \frac{d_2}{\alpha f_2} - \lambda \\ \vdots \\ \vdots \\ \frac{d_N}{f_N} - \lambda \end{bmatrix} - R_f^{-1}(\underline{f} - \bar{\underline{f}}) = \underline{0} \quad (10)$$

From equation (10), we know if the norm of $R_f(\|R_f\|)$ is very large, then

$$\underline{f}_{MAP} \approx \underline{d} = \underline{f}_{ML}$$

where \underline{d} is an observation data vector and \underline{f}_{ML} is the maximum likelihood estimates vector. In the blurred image case, the \underline{f}_{ML} is the inverse solution instead of the observation data.

On the other hand, if the norm of R_f is very small then

$$\underline{f}_{MAP} = \bar{\underline{f}}$$

where $\bar{\underline{f}}$ is an a priori non-stationary mean of the image.

Therefore, \underline{f}_{MAP} is a method which tries to move the solution of \underline{f} from the a priori non-stationary mean \underline{f} to a maximum likelihood estimate \underline{f} . Here R_f is a measure of our confidence in the non-stationary mean \underline{f} and maximum likelihood estimate as a solution to the restoration problem. Equation (10) appears very simple, but the complexity of the estimate implementation depends heavily on the structure of the R_f covariance matrix. Thus we will discuss in the two following sections methods of implementing equation (10). One assumes R_f is an identity matrix, and the other assumes that R_f is a Markovian matrix.

MAP Estimate Implementation with an A Priori Covariance Matrix of Image - Identity Covariance Matrix

For simplicity, we assume $R = \sigma_f^2 I$, thus each pixel of the picture is uncorrelated. Indeed, each pixel is highly correlated with its neighbors in realistic picture data, so we treat this assumption in the next section. From Equation (10) and $R = \sigma_f^2 I$ then we have

$$\begin{bmatrix} \frac{d_1}{\alpha f_1} - 1 \\ \frac{d_2}{\alpha f_2} - 1 \\ \vdots \\ \frac{d_N}{\alpha f_N} - 1 \end{bmatrix} - \frac{1}{\sigma_f^2} \begin{bmatrix} f_1 - \bar{f}_1 \\ f_2 - \bar{f}_2 \\ \vdots \\ f_N - \bar{f}_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (11)$$

From equation (11), we see that the MAP estimate becomes a very simple point process instead of vector process because there is no coupling between pixels. Hence we can get a closed form solution

$$f_i = \frac{(\bar{f}_i - \lambda \sigma_f^2) \pm \sqrt{(\bar{f}_i - \lambda \sigma_f^2)^2 + 4 \lambda \sigma_f^2 d_i}}{2} \quad (12)$$

We take the positive root because intensity is always non negative

Implementation

We need observation data in simulations and also must estimate the variance σ_f^2 and non-stationary mean \bar{f} from the observation data in order to implement equation (12) because the observation data is the only available data in practice.

The observation data are photon counts with some amplification gain . The photon count is simulated from an original picture (256x256) through a Poisson random generator. The Poisson random number generator used is a very fast, accurate algorithm which will be described in detail elsewhere. This algorithm is also available in the IMSL subroutine package (GGPOSH).

The non-stationary mean is estimated by a 1-dimensional moving average on 11 points of observation data and its variance is estimated by an unbiased estimate of the population variance.

The restored pictures are shown in Fig. 4 for different $(S/N)_{rms}$. Where $(S/N)_{rms}$ = Mean r.m.s. Signal to Noise Ratio. The upper left picture of each picture is the ideal picture. The upper right picture of each picture is the Poisson noisy picture. The lower left picture of each picture is an estimated non-stationary mean picture. The lower right picture of each figure is the MAP restored image. From Fig. 4 it can be seen that the restored pictures are improved compared to the noisy pictures.

MAP Estimate Implementation With An A Priori Covariance Matrix Of The Image - Markovian Covariance Matrix

In this section, we assume R_f is Markovian covariance matrix with correlation coefficient ρ . The Markovian covariance matrix is a very good approximation for real image signals. It is hoped that this more complicated model will produce better results.

If R_f is Markovian covariance matrix

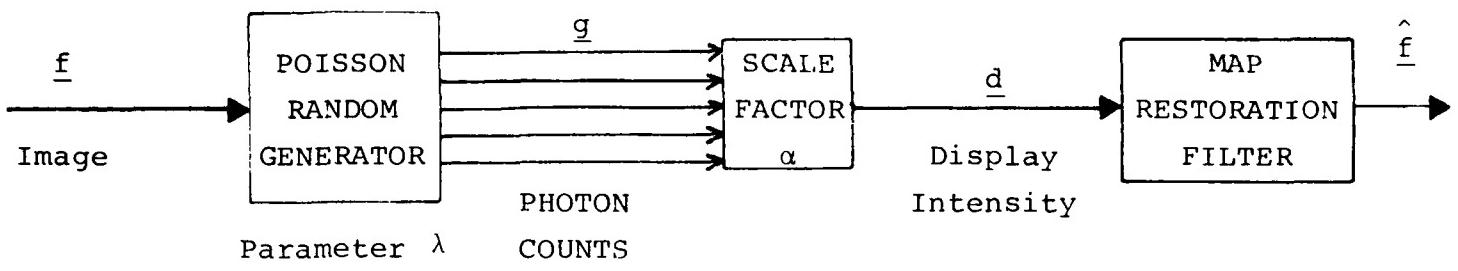


Fig. 3 Array Counter model for MAP Estimate

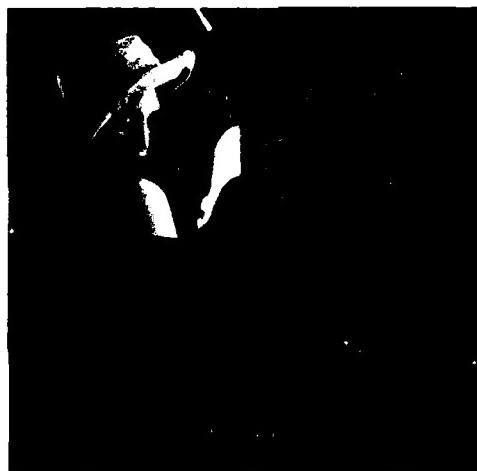


Fig. 4a $(S/N)_{rms} = \sqrt{2.5}$ Restored Picture



Fig. 4b Restored Picture for
 $(S/N)_{rms} = \sqrt{5}$



Fig. 4c Restored Picture for
 $(S/N)_{rms} = \sqrt{10}$

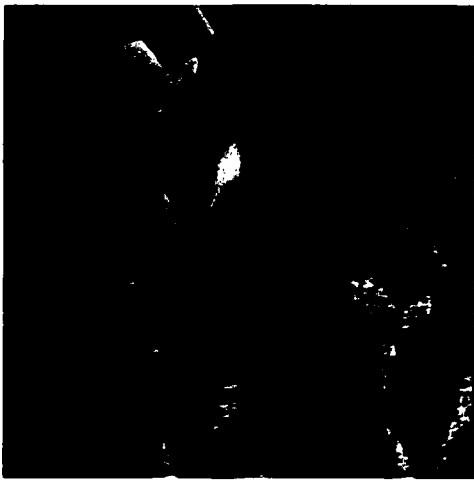


Fig. 5a Restored Picture for
overlapped-Add method



Fig. 5b Restored Picture for
overlapped-Save method

$$R_f = \sigma_f^2 \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & \dots & \rho^{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \dots & 1 \end{bmatrix} \quad (13a)$$

where $|\rho| < 1$. ρ is correlation coefficient between pixels
then

$$R_f^{-1} = \gamma \begin{bmatrix} \frac{1}{1+\rho^2} & -\beta & & & \\ -\beta & 1 & -\beta & & \\ & -\beta & 1 & -\beta & \\ & & -\beta & \frac{1}{1+\rho^2} & \end{bmatrix} \quad (13b)$$

where

$$\gamma = \frac{1+\rho^2}{1-\rho^2} \cdot \frac{1}{\sigma_f^2}$$

$$\beta = \frac{\rho}{1+\rho^2} \quad |\rho| < 1 \text{ and } |\beta| < \frac{1}{2}$$

Substituting equation (13b) into equation (10) we get three types of equations with N unknowns.

$$\left(\frac{d_1}{\alpha f_1} - \lambda\right) - \frac{\gamma}{1+\rho^2} (f_1 - \bar{f}_1) + \beta \gamma (f_2 - \bar{f}_2) = 0 \quad (8a)$$

$$\begin{aligned} \left(\frac{d_i}{\alpha f_i} - \lambda\right) + \beta \gamma (f_{i-1} - \bar{f}_{i-1}) - \gamma (f_i - \bar{f}_i) + \beta \gamma (f_{i+1} - \bar{f}_{i+1}) &= 0 \\ i = 2, 3, \dots, N-1 \end{aligned} \quad (8b)$$

$$\left(\frac{d_N}{\alpha f_N} - \lambda\right) + \beta \gamma (f_{N-1} - \bar{f}_{N-1}) - \frac{\gamma}{1+\rho^2} (f_N - \bar{f}_N) = 0 \quad (8c)$$

These N equations are nonlinear.

Due to the larger dimensionality and nonlinearity of the above system equations, ordinary linear signal processing techniques are of no use and the usual fast algorithms of linear signal processing cannot be used. The following strategy has been adopted for solution: First, we use a suboptimal sectioning method in order to reduce the dimensionality of the equations. We use two sectioning techniques [9]. One is the overlapped-add method (suitable for linear case), the other is overlapped-save method (suitable for linear or nonlinear cases). Second, we are seeking a good numerical method to solve the nonlinear equations. After trying several techniques for nonlinear equations, we have found that the Newton-Raphson iterative method is best. This method converges very fast in about 3 to 4 iterative steps. The detail of applying the Newton-Raphson iterative method to MAP estimation will be described in forthcoming reports.

Implementation

The estimates of the variance σ_f^2 and the non-stationary mean f are done by the same technique as in the last section. We have tried Newton-Raphson methods in both sectioning techniques (overlapped-add method and overlapped-save method). The results are shown as Fig. 5 for $(S/N)_{rms} = 6$ db. Fig. 5a uses the overlapped-add method and Fig. 5b uses the overlapped-save method. Of course, the overlapped-save method requires much more computing time than that of the overlapped-add method. However, from the Fig. 5 results, we can conclude that the two restored picture are perceptually the same. Thus, all the following processed pictures use the overlapped-add sectioning method. The restored pictures are shown in Fig. 6 for different $(S/N)_{rms}$ and $\rho=0$, $\rho=0.95$.

A Local Adaptive Processing

Because the MAP estimate contains an a priori non-stationary mean term and maximum likelihood term based on the local non-stationary statistical properties of the image, the local MAP estimate of the i th section is

$$Q_i * (\text{ML Term})_i + (1-Q_i) * (\text{A priori term}) = 0$$
$$i = 1, 2, \dots, L \quad L = \text{Total \# of sections}$$

Q_i can be adaptively varied in different sections depending on local statistical properties such as $\sigma_{f_i}^2$, and it is expected that this will improve the restored picture. For demonstration and simplicity, we have set $Q_i = Q_j = Q$ for all sections and the resulting pictures are shown as in Fig. 7.

From Fig. 7 we still can see that the restoration quality of Fig. 7a, Fig. 7b is better than that of Fig. 7c, Fig. 7d, even though Fig. 7a and Fig. 7b are globally weighted. (i.e. global adaptive processing).



Fig. 6a Restored Picture for
 $(S/N)_{rms} = \sqrt{2.5}$ $\rho = 0$
with Newton-Raphson Method



Fig. 6b Restored Picture for
 $(S/N)_{rms} = \sqrt{2.5}$ $\rho = 0.95$
with Newton-Raphson Method



Fig. 6c Restored Picture for
 $(S/N)_{rms} = \sqrt{5}$ $\rho = 0$
with Newton-Raphson Method



Fig. 6d Restored Picture for
 $(S/N)_{rms} = \sqrt{5}$ $\rho = 0.95$
with Newton-Raphson Method

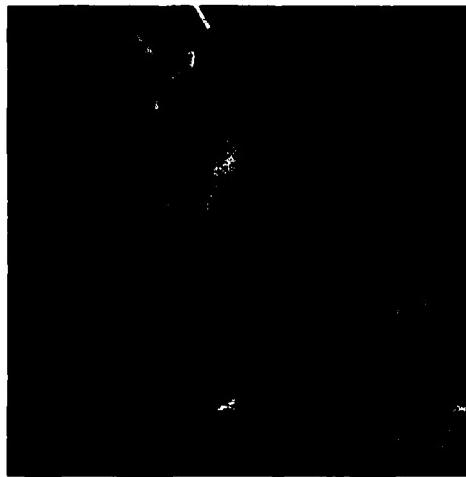


Fig. 6e Restored Picture for
 $(S/N)_{rms} = \sqrt{10}$ $\rho=0$
with Newton-Raphson Method

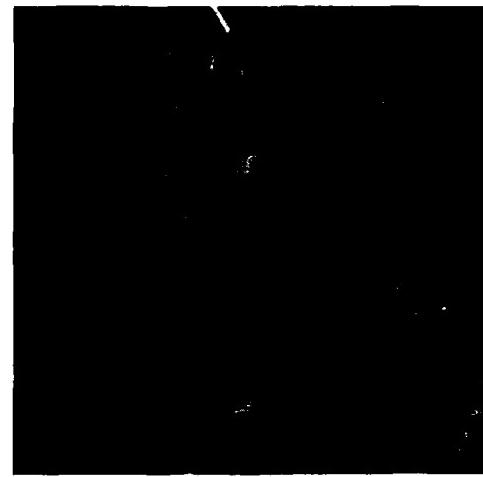


Fig. 6f Restored Picture for
 $(S/N)_{rms} = \sqrt{10}$ $\rho=0$
with Newton-Raphson Method



Fig. 7a Restored Picture for
 $(S/N)_{rms} = \sqrt{5}$, $\rho = 0$, $Q = 0.8$



Fig. 7b Restored Picture for
 $(S/N)_{rms} = \sqrt{5}$, $\rho = 0.9$, $Q = 0.8$



Fig. 7c Restored Picture for
 $(S/N)_{rms} = \sqrt{5}$, $\rho = 0$, $Q = 0.5$



Fig. 7d Restored Picture for
 $(S/N)_{rms} = \sqrt{5}$, $\rho = 0$, $Q = 0.5$

Conclusion

From this work, we have found that the estimated non-stationary mean carries most of the information in MAP estimation, and that the covariance carries much less information. However, the variance is a very important weighting factor in sectioning suboptimal MAP estimation because the local adaptive MAP processing method is very much dependent on the local non-stationary variance. So far, we have achieved very good preliminary results and a solid framework of MAP estimation in a Poisson Model. These results will encourage us to further investigate in blurring cases, in the local adaptive processing method and in space-variant blurring cases.

References

1. William K. Pratt, Digital Image Processing, Wiley-Interscience, 1978.
2. H.C. Andrews and B.R. Hunt, Digital Image Restoration, Prentice-Hall, 1977.
3. H.C. Andrews, Tutorial and selected papers in Digital Image Processing, IEEE Computer Society, 1978.
4. Harry L. Van Trees, Detection Estimation, and Modulation Theory, Part I, Wiley, 1968.
5. Harry L. Van Trees, Detection Estimation, and Modulation Theory, Part III, Radar-Sonar Signal Processing and Gaussian Signals in Noise, Wiley, 1971.
6. Ralph Deutsch, Estimation Theory, Prentice-Hall, 1965.

7. John B. Thomas, An Introduction to Statistical Communication Theory, Wiley, 1968.
8. B.R. Hunt, "Bayesian Methods in Nonlinear Digital Image Restoration", IEEE Transactions on Computer, March 1977.
9. H.J. Trussel and B.R. Hunt, "Sectioned Methods for Image Restoration", IEEE Transactions on ASSP, April 1978.
10. H.J. Trussel, "Notes on Linear Image Restoration by Maximizing the A Posteriori Probability", IEEE Transactions on ASSP, April 1978.
11. L.R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, 1975.
12. J.W. Goodman and J.F. Belsher, "Fundamental Limitations in Linear Invariant Restoration of Atmospherically Degraded Images", Proceeding of SPIE 75, 1976.
13. J.W. Goodman, "Precompensation and Postcompensation of Photon Limited Degraded Images", RADC-TR-76-382 ARPA Report, Dec. 1976.
14. J.W. Goodman, Statistical Optics Note 1976.
15. James J. Burke, "Estimating Objects from their Blurred and Graying Images", International Optical Computing Conference, April 1975.
16. B. Carnahan, H.A. Luther and J.O. Wilkes, Applied Numerical Methods, John Wiley and Sons, 1969.

17. Donald L. Snyder, Random Point Processes, John Wiley and Sons, 1975.

18. A.D. Whalen, Detection of Signal in Noise, Academic Press, 1971.

19. A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, 1965.

20. T.S. Huang, Picture Processing and Digital Filtering, Springer-Verlag, 1975.

3.3 Computer Hologram Interpolation with the DFT

Chung-Kai Hsueh and Alexander A. Sawchuk

Introduction

When making a computer generated hologram we usually embed the object in a zero array in order to further separate the reconstructions in adjacent orders. If the hologram is used as a filter then the size of the zero array may be very large compared to the impulse response in order to leave enough space for the object to be filtered. This process is actually an oversampling in the transform domain and does not provide any further information at all. If certain intermediate processes are to be used (for example an iterative phase coding method [1]), then the use of a large zero array not only wastes computation time but also requires more core as working space.

Thus it is desirable to work on the original object array of size N^2 and obtain a reconstruction of size $(MN)^2$ with the object embedded in a zero array. Note that

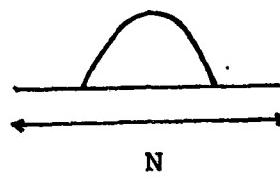
embedding the object in a zero array does not change the original sampling values in the transform domain. It simply fills in more points between the original sampling points. Since the zero array does not provide any new information the new points in the transform domain should be determined by the original sampling points. This suggests that our goal can be achieved by certain interpolation in the transform domain. One intuitive way is to use the translation property of the Fourier transform [2]. The object can be multiplied by a linear phase in order to get a translated Fourier transform. However, the reconstruction from this method has an object slightly off-axis. We found that instead of multiplying the object by linear phase, a modulated linear phase will produce the desired result. We also realized that this new method is actually a sinc function interpolation which has been claimed to be time consuming and not rigorous because of the infinite number of terms involved.

In section II we present our new method and show how it works. In section III we discuss its relationship with sinc function interpolation and show that they are indentical. Finally, in section IV, computation time and core requirements are compared and section V is the conclusion.

Theory

To simplify the mathematics we use a one dimensional object in this section. Extension to the two dimensional case is straightforward.

Suppose the original object has length N as shown in Fig.1(a). The input sequences, h_n $n=0,1,\dots,N-1$, are shown in Fig.1(b). Both figures are shown in continuous form although they are actually sampled. Now we want to make a



(a) Object



(b) Input sequences to DTF, h_n , $n = 0, \dots, N-1$

Figure 1

hologram such that the reconstruction has the same object surrounded by an MN zero array.

Step 1

Form $H_k^{(m)}$ as follows:

$$H_k^{(m)} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} h_n^{(m)} e^{-j2\pi kn/N} \quad k=0, \dots, N-1 \\ m=0, \dots, M-1 \quad (1)$$

where

$$h_n^{(m)} = h_n e^{-j2\pi m(n+\frac{N}{2}) \bmod N / MN} \quad (2)$$

Step 2

Let $\tilde{H}_k^{(m)} = H_k^{(m)} e^{j\pi m/M}$

$$\quad k=0, \dots, N-1 \\ m=0, \dots, M-1 \quad (3)$$

Step 3

Form a new sequence $P_\ell \quad \ell=0, \dots, MN-1$

$$P_\ell = \tilde{H}_{(\ell-m)/M}^{(m)} \quad \text{if } \text{Mod}(\ell, M) = m \quad (4)$$

Then P_ℓ has the desired property.

Proof:

Let Q_n be the inverse discrete Fourier transform of P_ℓ .

$$\begin{aligned} Q_n &= (\text{DFT})^{-1} P_\ell \\ &= \frac{1}{\sqrt{MN}} \sum_{\ell=0}^{MN-1} P_\ell e^{j2\pi\ell n/MN} \end{aligned} \quad (5)$$

Substituting Eqs. (4) and (3) we have

$$\begin{aligned} Q_n &= \sum_{m=0}^{M-1} \frac{1}{\sqrt{MN}} \sum_{k=0}^{N-1} \tilde{H}_k^{(m)} e^{j2\pi(Mk+m)n/MN} \\ &= \sum_{m=0}^{M-1} \frac{e^{j\pi m/M}}{\sqrt{MN}} \sum_{k=0}^{N-1} \tilde{H}_k^{(m)} e^{j2\pi(Mk+m)n/MN} \\ &= \sum_{m=0}^{M-1} \frac{e^{j\pi m/M}}{\sqrt{M}} \left[\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \tilde{H}_k^{(m)} e^{j2\pi kn/N} \right] e^{j2\pi mn/MN} \end{aligned} \quad (6)$$

$$\text{Let } n = pN + n' \quad n' = 0, \dots, N-1 \quad p = 0, \dots, M-1 \quad (7)$$

$$Q_n = Q_{pN+n'} \\ = \sum_{m=0}^{M-1} \frac{e^{j\pi m/M}}{\sqrt{M}} \left[\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} h_k^{(m)} e^{j2\pi k(pN+n')/N} \right] \\ \cdot e^{j2\pi m(pN+n')/MN} \quad (8)$$

$$= \sum_{m=0}^{M-1} \frac{e^{j\pi m/M}}{\sqrt{M}} \left[\frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} h_k^{(m)} e^{j2\pi kn'/N} \right] \\ \cdot e^{j2\pi mn'/MN} e^{j2\pi mp/M}$$

Using the inverse transform relationship of Eq.(1) and Eq.(2) we get

$$Q_{pN+n'} = \sum_{m=0}^{M-1} \frac{e^{j\pi m/M}}{\sqrt{M}} h_{n'}^{(m)} e^{j2\pi mn'/MN} e^{j2\pi mp/M} \\ = \sum_{m=0}^{M-1} \frac{e^{j\pi m/M}}{\sqrt{M}} \left[h_{n'}, e^{-j2\pi m(n'+N/2) \bmod N / MN} \right] \\ \cdot e^{j2\pi mn'/MN} e^{j2\pi mp/M} \quad (9)$$

Let us consider two cases of n'

$$(I) \quad n' = 0, \dots, \frac{N}{2} - 1$$

$$Q_{pN+n'} = \sum_{m=0}^{M-1} \frac{e^{j\pi m/M}}{\sqrt{M}} \left[h_n e^{-j2\pi mn'/MN} e^{-j\pi m/M} \right] \\ \cdot e^{j2\pi mn'/MN} e^{j2\pi mp/M} \quad (10)$$

$$= \frac{1}{\sqrt{M}} h_n \sum_{m=0}^{M-1} e^{j2\pi mp/M}$$

Since

$$\sum_{m=0}^{M-1} x^m = \frac{1-x^M}{1-x} \quad (11)$$

we have

$$s = \sum_{m=0}^{M-1} e^{j2\pi mp/M} = \frac{1-e^{j2\pi mp}}{1-e^{j2\pi mp/M}} \quad (12)$$

If $p = 1, 2, \dots, M-1$

$$S = \frac{1-1}{1-e^{j2\pi mp/M}} = 0 \quad (13)$$

If $p=0$, we substitute $p=0$ in the original expression

$$S = \sum_{m=0}^{M-1} 1 = M \quad (14)$$

Therefore Eq. (10) becomes

$$Q_{pN+n'} = \begin{cases} \sqrt{M} h_n, & p = 0 \\ 0 & p = 1, 2, \dots, I-1 \end{cases} \quad (15)$$

$$(II) \quad n' = \frac{N}{2}, \dots, N-1$$

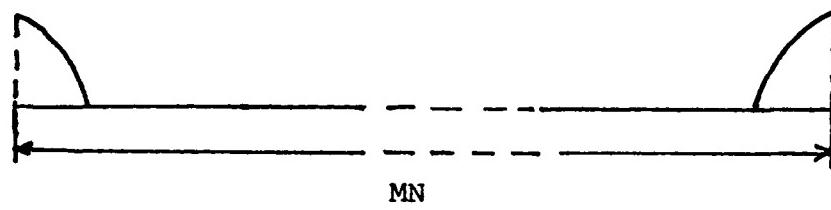
$$Q_{pN+n'} = \sum_{m=0}^{M-1} \frac{e^{j\pi m/M}}{\sqrt{M}} \left[h_n e^{-j2\pi m(n'-N/2)/MN} \right] \\ \cdot e^{j2\pi mn'/MN} e^{j2\pi mp/M}$$

$$\begin{aligned}
 &= \frac{1}{\sqrt{M}} h_n \sum_{m=0}^{M-1} e^{j\pi m/M} e^{-j2\pi mn'/MN} e^{j\pi m/M} \\
 &\quad \cdot e^{j2\pi mn'/MN} e^{j2\pi mp/M} \\
 &= \frac{1}{\sqrt{M}} h_n \sum_{m=0}^{M-1} e^{j2\pi m(1+p)/M} \\
 &= \begin{cases} \sqrt{M} h_n, & p = I-1 \\ 0, & p = 0, 1, \dots, I-2 \end{cases} \tag{16}
 \end{aligned}$$

Figure 2 shows the sequences Q_n and the reconstruction from the hologram. Figure 3 further shows the computer simulation of a two dimensional case. Figure 3(a) is the original object (letter 'P') and (b) is the reconstruction when the array is expanded by four times in each direction. Note that if the holograms are of the same size (before and after interpolation), then the object in (b) is of the same size as the object in (a) except that it is surrounded by a zero array.

Relationship with Sinc Interpolation

Let us consider the following Fourier transform pair $h(x) \leftrightarrow H(f)$ (Fig.4) where $h(x)$ is space-limited (width X) and $H(f)$ is not band-limited although $|H(f)|$ may be very small when $|f|$ goes beyond a certain frequency. Note that the word 'band-limited' used here has a slightly different meaning from its common use, but is self-explained in Figure 4. Suppose $h(x)$ is sampled at intervals ΔX , then aliasing error occurs due to the fact that no Nyquist rate



(a) Output sequences Q_n , $n = 0, \dots, MN-1$



(b) Reconstruction

Figure 2

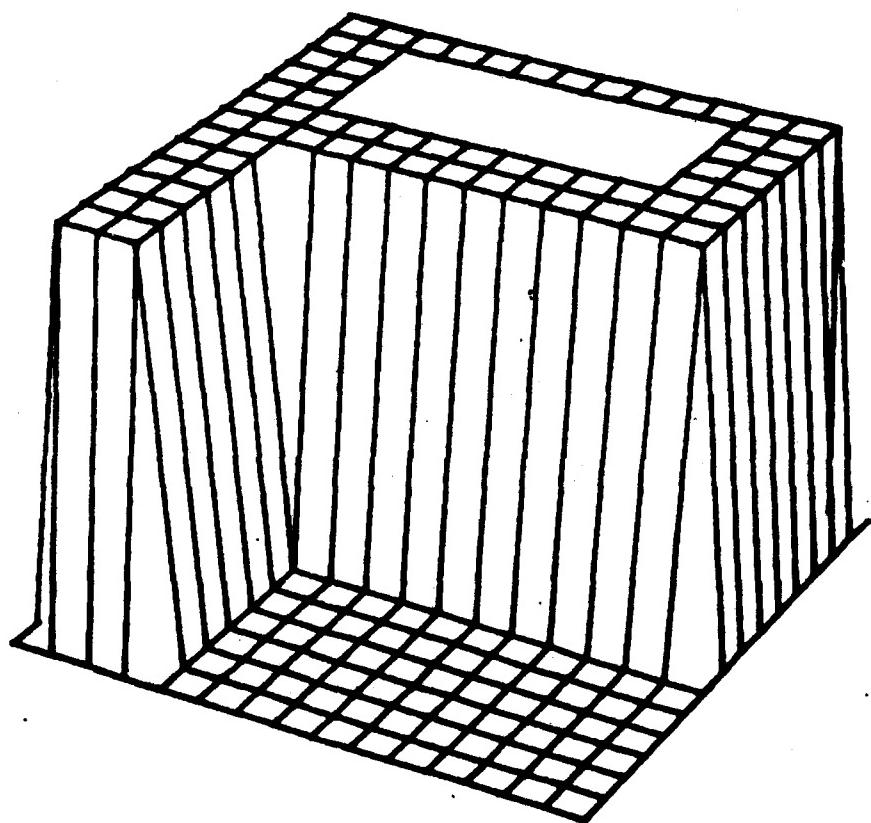


Fig. 3 (a) Two dimensional object (letter 'P')

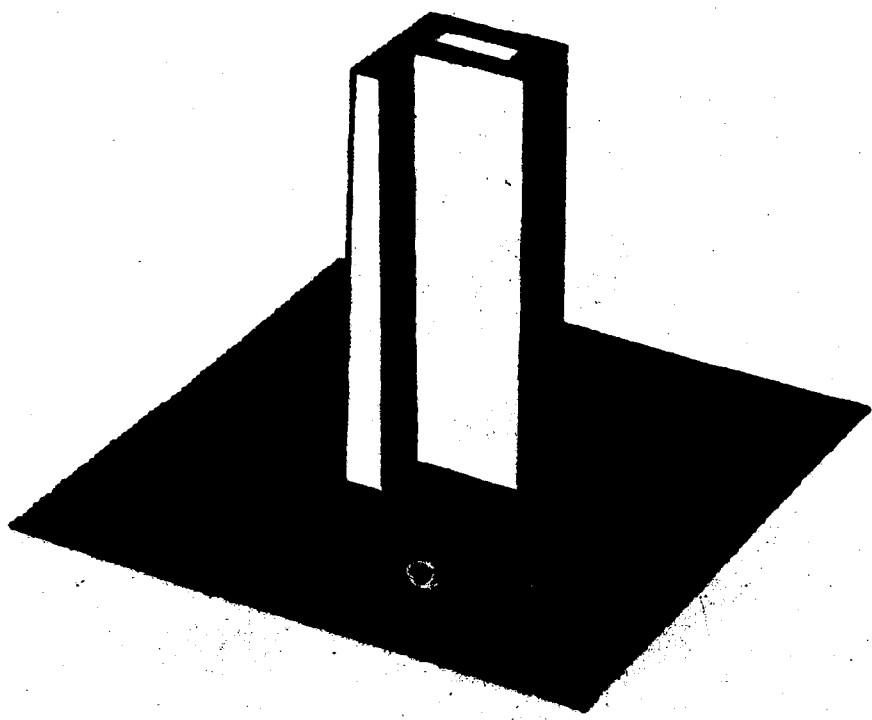


Fig. 3(b) Reconstruction ($M = 4$)

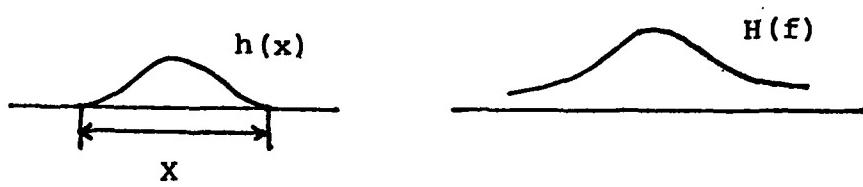


Figure 4. Fourier transform pair $h(x) \Leftrightarrow H(f)$

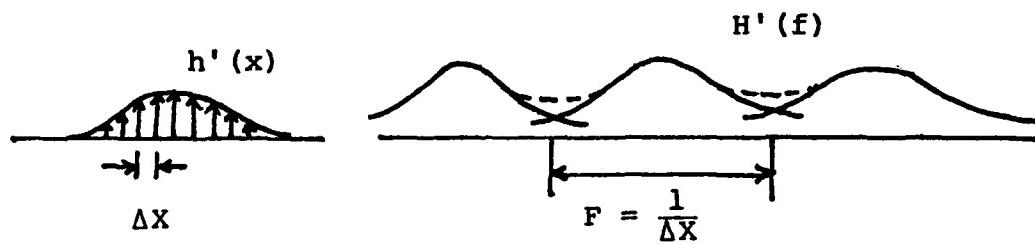


Figure 5. Fourier transform pair $h'(x) \Leftrightarrow H'(f)$
where $h'(x)$ is the sampled version
of $h(x)$

exists. Figure 5 shows the sampled function $h(x)$ and its Fourier transform $H'(f)$. The period of $H'(f)$ is given by

$$F = \frac{1}{\Delta X} \quad (17)$$

and there are N samples in $h'(x)$.

When using the discrete Fourier transform, we have to sample $H'(f)$ for the same number of points. The interval is then given by

$$\Delta F = \frac{F}{N} = \frac{1}{N\Delta X} = \frac{1}{X} \quad (18)$$

Figure 6 shows the new transform pair $h''(x) \leftrightarrow H''(f)$ and one period from each of them are what we actually get from the discrete Fourier transform.

If we use continuous sinc function interpolation for $H''(f)$ then we are equivalently filtering $h''(x)$ by a rect function and we get back to the relationship in Fig.5. Now if we sample $H'(f)$ at the interval of $\frac{1}{MX}$ instead of $\frac{1}{X}$, then we get a new transform pair $\tilde{h}''(x) \leftrightarrow \tilde{H}''(f)$ as shown in Fig.7. All of the MN points are the result of the sinc interpolation since they are the sampled values of the continuous sinc interpolation. Notice that $\tilde{h}''(x)$ is the sampled $h(x)$ embedded in a zero array of size MN . This is exactly the same result obtained from the method we proposed in section II. Since the Fourier transform uniquely determines the relationship between the object and the Fourier transform we can conclude that our method actually does the sinc function interpolation on finite points. It is also noted that by embedding the object in a zero array and performing Fourier transform we also perform the sinc

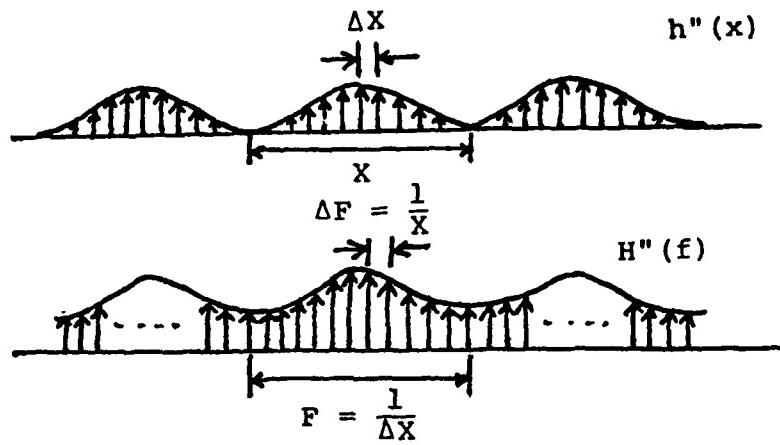


Figure 6. Fourier transform pair $h''(x) \Leftrightarrow H''(f)$ where both $h(x)$ and $H(f)$ are sampled.

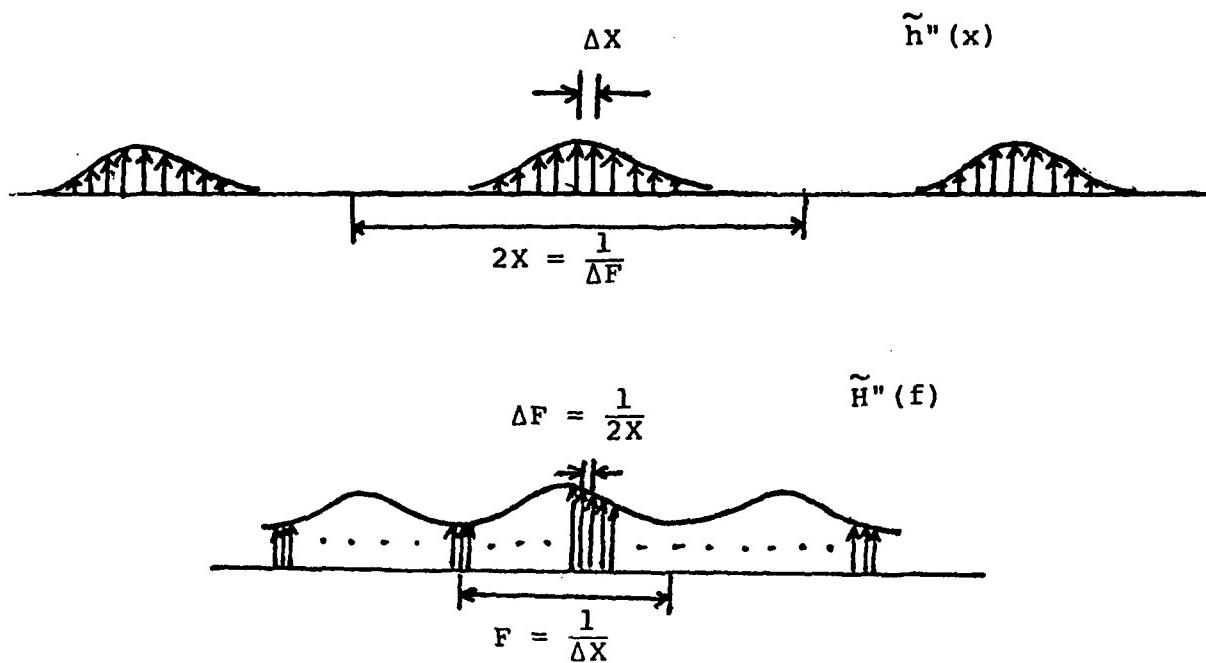


Figure 7. Fourier transform pair $\tilde{h}''(x) \Leftrightarrow \tilde{H}''(f)$ where $H(f)$ is sampled finer. ($M=2$)

interpolation. However, it requires more core as working space as discussed in the next section.

Comparisons

To illustrate the advantage of the new method let us consider the case when array is gone through the space-transform iterative phase coding method [3,4] for n times. The original array size is NxN and is to be put into a MNxMN zero array. Here we assume that M,N are power of 2.

When we use $(MN)^2$ points directly we need CPU time to do the Fourier transform (t_1) and the conversion between polar coordinate and Cartesian coordinate (t'_1).

$$\begin{aligned} t_1 &= (2n-1)[(MN)^2 \log_2 (MN)^2] \\ &= 2(2n-1)M^2 N^2 \log_2 (MN) \end{aligned} \quad (19)$$

Where 2 accounts for the two Fourier transforms required in each iteration and -1 accounts for the fact that we need only one Fourier transform in the first iteration.

To compute t'_1 , let us assume t_{pc} and t_{cp} be the time required for one component to do conversion from polar coordinate to Cartesian coordinate and vice versa. Also assume that the original object is in polar form.

Then

$$\begin{aligned} t'_1 &= (2n-1)(MN)^2(t_{pc} + t_{cp}) \\ &= (2n-1)M^2 N^2 t_0 \end{aligned} \quad (20)$$

where

$$t_0 = t_{pc} + t_{cp} \quad (21)$$

When using the new method we need CPU time for the iterative method (t_{21}), interpolation (t_{22}) and conversion (t'_2).

$$\begin{aligned} t_{21} &= (2n-1)N^2 \log_2 N^2 \\ &= 2(2n-1)N^2 \log_2 N \end{aligned} \quad (22)$$

To interpolate the points between the original points we have to go through Eqs. (1)-(4). The multiplication of an exponential term is simply an addition in phase and can be neglected. However, in Eq. (2) we do need 2 multiplications for each component and therefore we get

$$\begin{aligned} t_{22} &= (M^2-1)(N^2 \log_2 N^2 + 2N^2) \\ &= (M^2-1)2N^2(\log_2 N + 1) \end{aligned} \quad (23)$$

Let t_2 be the sum of t_{12} and t_{22} , we have

$$\begin{aligned} t_2 &= t_{12} + t_{22} \\ &= 2N^2[(2n-1)\log_2 N + (M^2-1)(\log_2 N + 1)] \end{aligned} \quad (24)$$

The conversion time t'_2 is given by

$$\begin{aligned} t'_2 &= (2n-1)N^2 t_0 + (M^2-1)N^2 t_0 \\ &= (2n+M^2-2)N^2 t_0 \end{aligned} \quad (25)$$

Let us consider some typical values of M, N , and n and see what is the difference between these two methods.

Suppose $n=20, N=32, M=4$, then

$$t_1 = 2N^2 \cdot 39 \cdot 16 \cdot 7 = 2N^2 \cdot 4368 \quad (26)$$

$$t_2 = 2N^2 [39 \cdot 5 + 15 \cdot 6] = 2N^2 \cdot 285 \quad (27)$$

$$\frac{t_1}{t_2} = \frac{4368}{285} \approx 15.3 \quad (28)$$

The saving of the CPU time for the non-conversion part is about a factor of 15.3.

$$t'_1 = 39 \cdot 16 N^2 t_0 = 624 N^2 t_0 \quad (29)$$

$$t'_2 = (40+16-2)N^2 t_0 = 54N^2 t_0 \quad (30)$$

$$\frac{t'_1}{t'_2} = \frac{624}{54} \approx 11.6 \quad (31)$$

Again the saving in data conversion is about a factor of 11.6.

For other larger values of n,M,N we can use the following approximations.

$$t_1 \approx 4nM^2 N^2 \log_2 N \quad (32)$$

$$\begin{aligned} t_2 &\approx 2N^2 [2n \log_2 N + M^2 \log_2 N] \\ &\approx 2N^2 (2n + M^2) \log_2 N \end{aligned} \quad (33)$$

$$t'_1 \approx 2nM^2 N^2 t_0 \quad (34)$$

$$t_2' \approx (2n+M^2)N^2 t_0 \quad (35)$$

$$\frac{t_1}{t_2} \approx \frac{t_1'}{t_2'} \approx \frac{2nM^2}{2n+M^2} \quad (36)$$

It is obvious from Eq. (36) that the new method always takes less CPU time for large n and M.

There is one other way to save CPU time in making the hologram mentioned above. That is to perform the iterative phase coding method on N^2 points then embed this result in a $(MN)^2$ zero array and perform Fourier transform. Let t_3 and t_3' be the CPU time required for Fourier transforming and data conversion, respectively.

$$\begin{aligned} t_3 &= (2n-1)N^2 \log_2 N^2 + (MN)^2 \log_2 (MN)^2 \\ &= 2(2n-1)N^2 \log_2 N + 2M^2 N^2 \log_2 (MN) \end{aligned} \quad (37)$$

Compared with t_2 in Eq. (24), t_3 is larger but approximately equal to t_2 . The conversion time t_3' is given by

$$t_3' = (2n-1)N^2 t_0 + (MN)^2 t_{cp} \quad (38)$$

t_3' is smaller than t_2' . However, this method requires large core to do the computation when M is large.

Conclusion

We have introduced a new method to perform interpolation which utilizes the translation property of the Fourier transform. This method has reduced the amount of CPU time and core size in making a computer generated hologram which reconstructs an object embedded in big zero array. We also showed that this method is a practical implementation of sinc function interpolation without any error caused by the truncation of the impulse response. Due to the property of performing sinc interpolation this method may find other applications in other areas.

References

1. C.K. Hsueh and A.A. Sawchuk, "Phase Coding for Optical Image Processing," USCIP Report 720, pp. 133-145, (Sept. 1976).
2. C. Chavel and J.P. Hugonin, "High Quality Computer Holograms: The Problem of Phase Representation," J. Opt. Soc. Am., 66, pp. 989-996, (1976).
3. J.P. Allebach, N.C. Gallagher, and B.Liu, "Aliasing Error in Digital Holography," Appl. Opt., 15, pp. 2183-2188 (1976).
4. C.K. Hsueh and A.A. Sawchuk, "Computer-Generated Double Phase Holograms," accepted for publication in December 1978 Applied Optics.

3.4 Spotlight S.A.R. Imaging Using RAT-SCAT Site Date

Peter Chuan

Introduction

It has been known [1], [5], [6] for more than a decade that the shape and size of perfectly conducting targets can be reconstructed by relating the target's three dimensional Fourier transform to the monostatic data collected for all carrier frequencies and target aspect angle.⁺ Two dimensional reconstruction of rotating targets [2] have however been more successfully demonstrated using Range-Doppler Principle [3], [10]. Two dimensional reconstruction not explicitly using the Range-Doppler Principle has also been used in radar astronomy to map planetary objects [4]. Two dimensional mapping techniques used in radar astronomy collects reflected data as a function of the aspect angle of the rotating body rather than as functions of time. Therefore the target angular velocity dependence of the data becomes invisible in the model. Despite this flurry of activity, it was not until recently that RAT-SCAT data was used for target mapping [7]. The imaging equations will be rederived, but here the effects of shadowing and reflectivity change will be considered.

The RATSCAT (RAdar Target SCATTERing Site) recording geometry is shown in Figure 1. The radar R transmits CW (continuous wave) sinusoidal signal of frequency f while the target T is fixed at a certain aspect angle θ_i . At the receiver , the signal directly from S and the signal reflected from T are mixed and integrated. This integrate and dump output forms one sample complex data which is recorded on a magnetic tape. The same process is repeated for all possible aspect angle θ_i , and all possible

⁺Aspect Angle is the angle between the line of sight of the radar beam and a coordinate axis frame fixed on the target. See fig. 2.

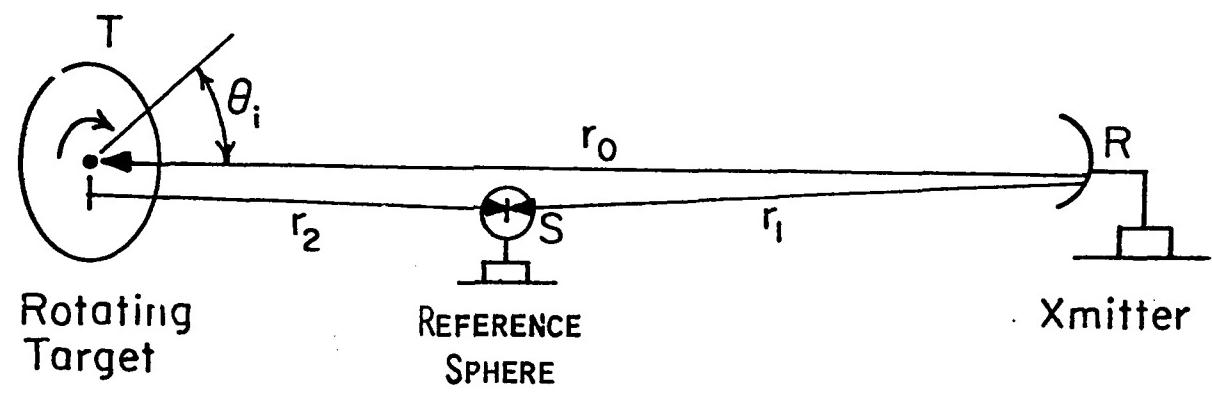


Fig. 1. RAT-SCAT Site Recording Geometry.

carrier frequencies f_k .

Formulation

A. Small Target Assumption

The basic assumption on the recording geometry is that the target T, reference sphere S, and radar transmitter R are all approximately collinear and non-collinear effects [7] will be negligible. Let

r_0 =distance between target center and
transmitter,

r_1 =distance between reference sphere and
transmitter,

r_2 =distance between target center and reference
sphere.

Lines of constant range as seen by the receiver are shown in Figure 2. Ideally, these lines should have no curvature and the maximum error suffered from this assumption is the well known range walking problem [8], [9].

$$\epsilon_r = \sqrt{a^2 + r_0^2} - r_0$$

$$\epsilon_r \cong \frac{a^2}{2r_0}$$

(Taylor's expansion)

$$< \rho_R$$

where a is half the maximum linear extent of the target and ρ_R is the desired linear target resolution. Therefore

$$a < \sqrt{2\rho_R r_0}$$

must be satisfied. In less exact terms, this is the small target assumption

$$a \ll r_0$$

B. Specular Assumption

The second fundamental assumption is that only specular radar returns are involved, since signal wavelength λ_k is much smaller than the curvature of the target surface. This is the fundamental assumption that will be used here. Other processes like creeping wave returns [10] and reverberative returns (multiple interval reflections within the target) are also known to contribute to radar return signal, but these considerations will be ignored here.

C. Consideration on Changing Reflectivity

Figure 3 shows the illumination pattern on the target. Let (ξ, η) be the coordinates of the target body and (x, y) be the coordinates of the ground reference frame. Both radar and receiver are on the x-axis.

Let a unit length normal vector $\underline{n}(\xi, \eta)$ be associated with each point (ξ, η) on the target surface. Define

$$\alpha_{\xi, \eta} = \text{angle between } \xi \text{ axis and } \underline{n}(\xi, \eta)$$

Then the reflectivity of the target is

$$\sigma(\xi, \eta; \theta_i) = \sigma_0(\xi, \eta) \cos(\alpha_{\xi, \eta} - \theta_i)$$

where $\sigma_0(\xi, \eta)$ is the maximum reflectivity of a target point over all θ_i .

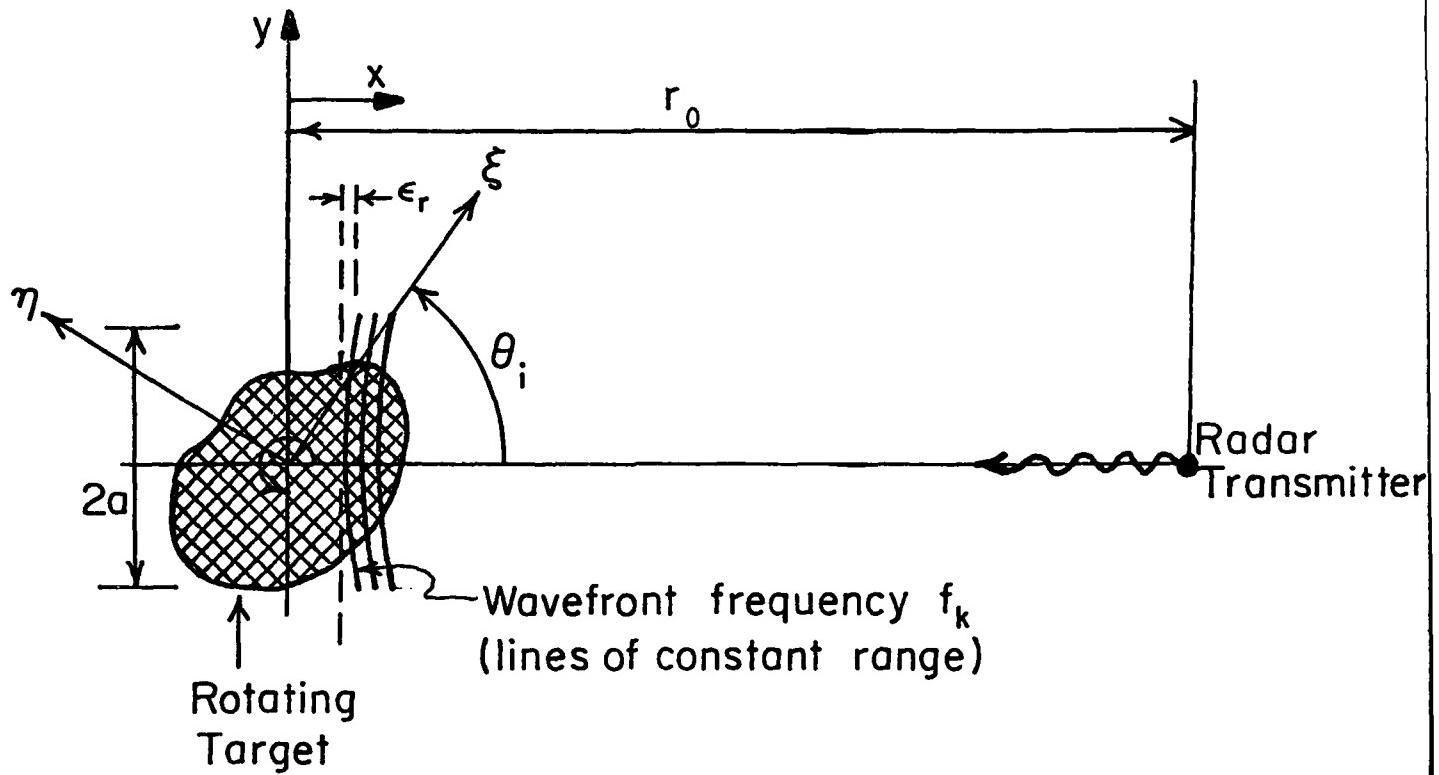


Fig. 2. Target coordinates and plane wavefront assumptions.

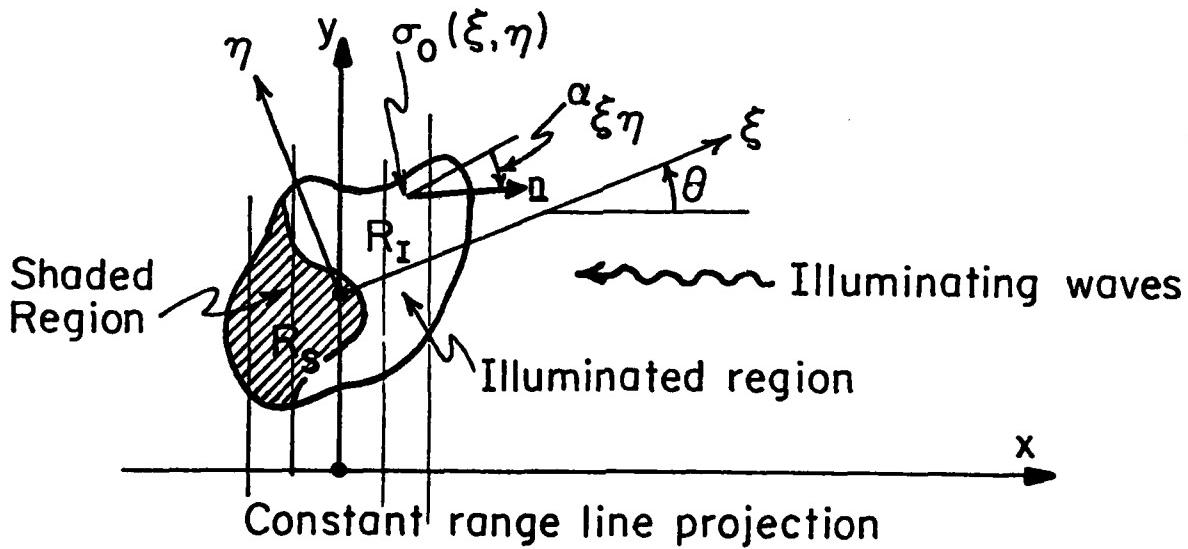


Fig. 3. Illumination pattern of the target at
(frequency, aspect angle) = (f_k, θ_i)

D. Representation of Recorded Data

Figure 3 also shows the illuminated region R_I and the shadow region R_S which is in the shadow. Let the sinusoidal wave transmitted at T be

$$f(t) = \cos(2\pi f_k t + \phi)$$

where ϕ is some unknown but fixed phase. From Appendix A, the recorded data at S is

$$D_I(i, k) = D e^{+j2\pi(\frac{2f_k}{c})(\frac{r_0 - r_1}{2})} z_I(\theta_i, f_k)$$

where

$$z_I(\theta_i, f_k) = \iint_{R_I} \sigma(\xi, n; \theta_i) e^{-j2\pi(\frac{2f_k}{c})(\xi \cos \theta_i + n \sin \theta_i)} d\xi dn.$$

and D is a complex constant. The extra phase factor $e^{j2\pi(\frac{2f_k}{c})(r_0 - r_1)}$ can be estimated and compensated for, thus we are left with $z_I(\theta_i, f_k)$.

$z_I(\theta_i, f_k)$ becomes the Fourier transform of the target reflectivity if two previous assumptions can be relaxed.

1. The reflectivity is independent of the aspect angle of the target;

$$\sigma(\xi, n; \theta_i) \approx \sigma_0(\xi, n)$$

2. There is no shadowing effect;

$$R_I = \text{entire target area.}$$

Then

$$z_I(\theta_i, f_k) = z(\theta, f) \Bigg|_{\begin{array}{l} \theta=\theta_i \\ f=\frac{2f_k}{c} \end{array}} \quad (1)$$

where

$$z(\theta, f) = \iint_{-\infty}^{\infty} \sigma_0(\xi, \eta) e^{-j2\pi f(\xi \cos \theta + \eta \sin \theta)} d\xi d\eta$$

Condition 1 holds when the target consists of a distribution of point scatterers. Condition 2 holds when the target is a convex object. Both conditions 1 and 2 were satisfied for the targets used by Walker [5] in his reconstruction and as a consequence, the recorded data were an exact Fourier transform of the target reflectivity. In practice, targets of interest usually do not satisfy these conditions and therefore direct Fourier transforming the data without further consideration may fail to produce good reconstructions.

Due to physical optics approximation and propagation problems, f_k is bounded below and due to device limitations f_k is bounded above. Therefore, the data $z_I(\theta_i, f_k)$ is bandlimited, or it is confined to a ring. This is shown in Figure 4. If only a small segment (see [7])

$$f_k \in [f_{\min}, f_{\max}]$$

$$\theta_i \in [\theta_n - \Delta\theta, \theta_n + \Delta\theta] \quad (\theta_n = \text{mean aspect angle})$$

is used to reconstruct the target, target resolution will suffer, but nevertheless both conditions 1 and 2 can be neglected without great error. First note that

HD-A134 989

IMAGE UNDERSTANDING RESEARCH(U) UNIVERSITY OF SOUTHERN
CALIFORNIA LOS ANGELES IMAGE PROCESSING INST

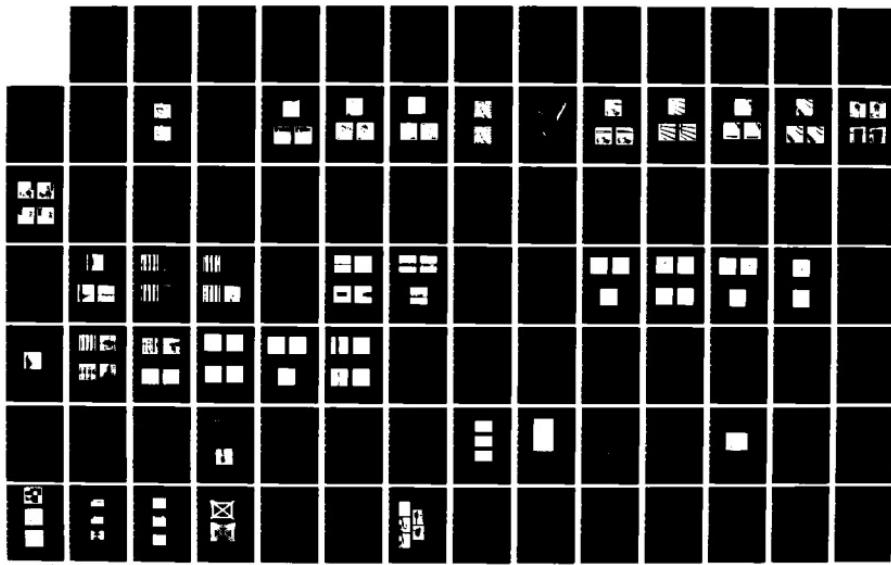
3/4

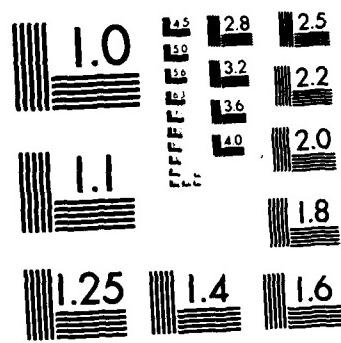
H C ANDREWS 30 SEP 78 USCIP1-840 F33615-76-C-1203

F/G 20/6

NL

UNCLASSIFIED





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

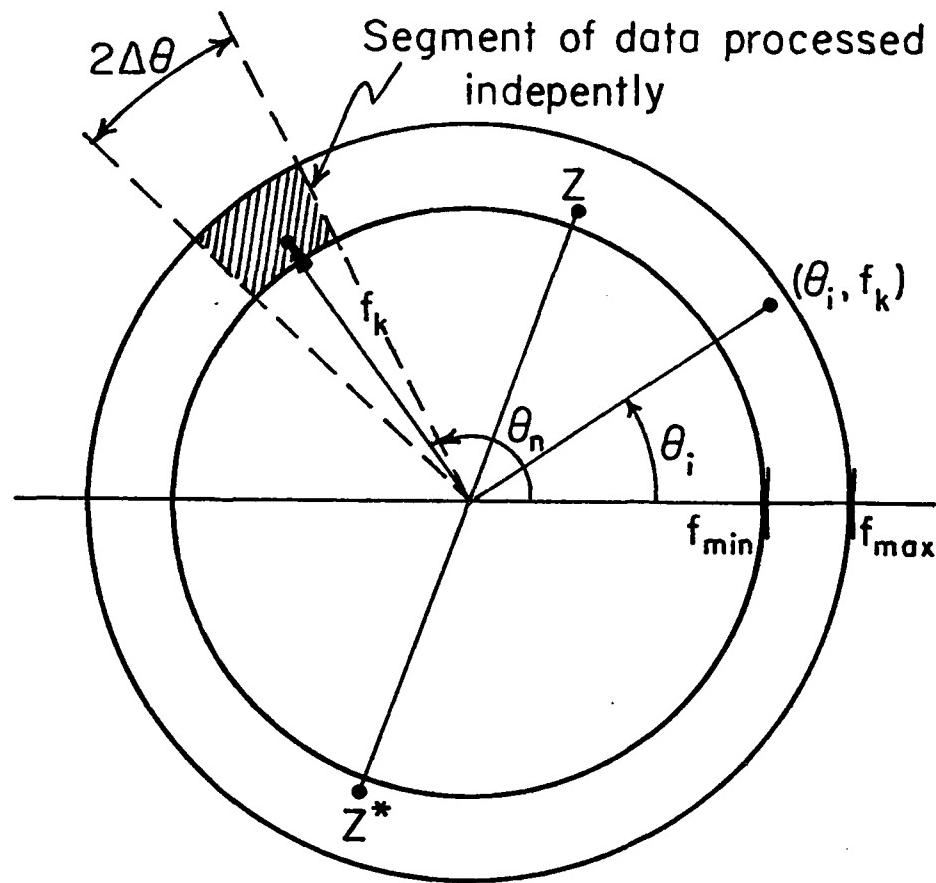


Fig. 4. 2-Dimensional representation of complex data $D_I(i,k)$ for the RAT-SCAT system. After compensating for the term $\exp[j2\pi(f_k/c)(r_0-r_1)]$, $D_I(i,k)$ is the F.T. of the target if:

1. the target reflectivity is independent of θ_i ;
2. there is no shadowing.

$$\begin{aligned}\sigma(\xi, \eta; \theta_i) &= \sigma_0(\xi, \eta) \cos(\alpha_{\xi, \eta} - \theta_i) \\ &\cong \sigma_0(\xi, \eta) \cos(\alpha_{\xi, \eta} - \theta_n)\end{aligned}$$

over $\theta_i \in [\theta_n - \Delta\theta, \theta_n + \Delta\theta]$, so that $\sigma(\xi, \eta; \theta_i)$ is approximately independent of θ_i . Also, because the target is now observed for only such a small angular extent ($2\Delta\theta$), the shape of R_I and R_S will remain essentially constant and hence shadowing effect poses no great problem. This was in fact the case for the segment processing method used by Chen [7].

Shadowing Effect

Since only data from the illuminated side of the object are available, one method to solve the shadowing problem is to combine two or more looks on the target. For better insight into the problem we will make use of the line projection integral of Equation (2) in Appendix A.

$$g_I(x, \theta_i) = \iint_{R_I} \sigma_0(\xi, \eta) \delta(\xi \cos \theta_i + \eta \sin \theta_i - x) d\xi d\eta$$

where we have assumed that the target reflectivity distribution is independent of θ_i . The projection slice theorem can then be used to rewrite the RAT-SCAT Data.

$$z_I(\theta_i, f_k) = \int_{-a}^a g_I(x, \theta_i) e^{-j2\pi(\frac{2f_k}{c})x} dx \quad (2)$$

where z_I is defined in (1) and a is half the maximum spatial extent of the target. Then

$$D_I(i,k) = D e^{+j2\pi(\frac{2f_k}{c})(r_0-r_1)} z_I(\theta_i, f_k)$$

For a given object and illumination angle θ_i , the object field can be partitioned into two non-overlapping regions R_I and R_S where

R_I represents target field illuminated
 R_S represents target field in the shadow.

Let $g_S(x, \theta_i) \triangleq \iint_{R_S} \sigma_O(\xi, \eta) \delta(\xi \cos \theta_i + \eta \sin \theta_i - x) d\xi d\eta$

Then

$$g_I(x; \theta_i + \pi) = \iint_{R'_I} \sigma_O(\xi, \eta) \delta(\xi \cos[\theta_i + \pi] + \eta \sin[\theta_i + \pi] - x') d\xi d\eta$$

Where R'_I is the region illuminated after rotating the target 180° . If $R'_I = R_S$ = the shadow region prior to rotating the target 180° (see Figure 5) then

$$g_I(x; \theta_i + \pi) = \iint_{R_S} \sigma_O(\xi, \eta) \delta(\xi \cos[\theta_i + \pi] + \eta \sin[\theta_i + \pi] - x') d\xi d\eta$$

Therefore

$$g_I(-x, \theta_i + \pi) = \iint_{R_S} \sigma_O(\xi, \eta) \delta(-\xi \cos \theta_i - \eta \sin \theta_i + x) d\xi d\eta \quad (3)$$

i.e.

$$g_I(-x, \theta_i + \pi) = g_S(x, \theta_i)$$

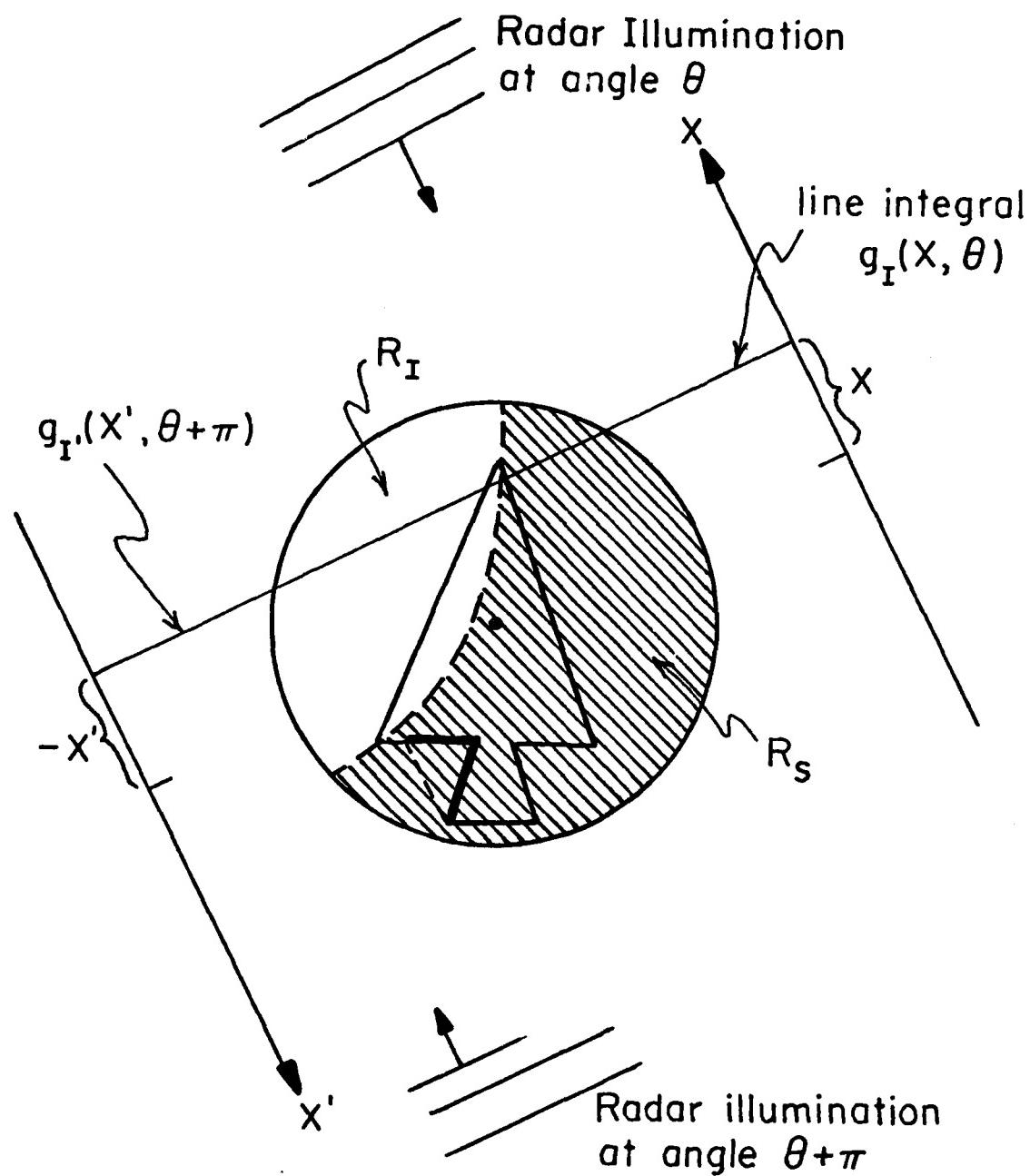


Fig. 5. Shadow and illuminated regions of the target field. The heavy lines show the "double" shadow part of the target which cannot be reached by the radar waves (because of specularity assumption) from both directions θ and $\theta+\pi$.

Thus, the line integral taken by rotating the target 180° is the "shadow" line integral. However, the desired line projection integral is

$$g(x, \theta_i) \triangleq \iint_{R_I^{URS}} \sigma_0(\xi, \eta) \delta(\xi \cos \theta_i + \eta \sin \theta_i - x) d\xi d\eta$$

$$g(x, \theta_i) = g_I(x, \theta_i) + g_S(x, \theta_i)$$

$$g(x, \theta_i) = g_I(x, \theta_i) + g_I(-x, \theta_i + \pi) \quad (4)$$

We have available only the data (after compensation for the extra phase factor)

$$z_I(\theta_i, f_k) = \int_{-a}^a g_I(x, \theta_i) e^{-j2\pi(\frac{2f_k}{c})x} dx$$

and we wish to get (from applying (4))

$$\begin{aligned} z(\theta_i, f_k) &= \int_{-a}^a g(x, \theta_i) e^{-j2\pi(\frac{2f_k}{c})x} dx \\ &= \int_{-a}^a g_I(x, \theta_i) e^{-j2\pi(\frac{2f_k}{c})x} dx + \int_{-a}^a g_I(-x, \theta_i + \pi) e^{-j2\pi(\frac{2f_k}{c})x} dx \\ &= z_I(\theta_i, f_k) + z_I(\theta_i + \pi, -f_k) \end{aligned}$$

But

$$z_I(\theta_i + \pi, -f_k) = z_I^*(\theta_i + \pi, f_k) \quad (5)$$

where * represents complex conjugate. Therefore

$$z(\theta_i, f_k) = z_I(\theta_i, f_k) + z_I^*(\theta_i + \pi, f_k) \quad (6)$$

and shadowing can be taken care of by recomputing the data with equation (6). We also notice from (6) that

$$\begin{aligned} z(\theta, -f) &= z_I(\theta, -f) + z_I^*(\theta + \pi, -f) \\ &= z_I^*(\theta, f) + z_I^*(\theta + \pi, -f) \\ &= [z_I(\theta, f) + z_I^*(\theta + \pi, -f)]^* \\ &= [z_I(\theta, f) + z_I^*(\theta + \pi, f)]^* \text{ Relation (5)} \\ &= z^*(\theta, f) \end{aligned}$$

$$z(\theta, -f) = z^*(\theta, f)$$

i.e. the data $z(\theta, f)$ has conjugate symmetry and reconstructing the target by taking the F.T. of $z(\theta, f)$ will give real numbers, which is consistent with intuition.

Finally, because the target is usually not convex in which case some regions in the target field will not be covered by both R_I and R_S and will consequently represent an error in the initial assumption that the entire target field is partitioned into R_S and R_I regions only. We will call this effect "double shadowing." Let

$$g_{DS}(x, \theta) \triangleq \iint_{R_{DS}} \sigma_0(\xi, \eta) \delta(\xi \cos \theta + \eta \sin \theta - x) d\xi d\eta$$

where R_{DS} is the double shadow region. Then ideally, the line integral is

$$g(x, \theta) = \iint_{R_I \cup R_S \cup R_{DS}} \sigma_0(\xi, \eta) \delta(\xi \cos \theta + \eta \sin \theta - x) d\xi d\eta$$

and the ideal phase corrected data is

$$z'(\theta_i, f_k) = z(\theta_i, f_k) + \epsilon(\theta_i, f_k)$$

where $\epsilon(\theta_i, f_k)$ is the error due to double shadowing

$$\epsilon(\theta_i, f_k) = \int_{-a}^a g_{DS}(x, \theta_i) e^{-j2\pi(\frac{2f_k}{c})x} dx$$

and $z(\theta_i, f_k)$ is defined in equation (6).

The next obvious step to correct $\epsilon(\theta_i, f_k)$ error is to infer $g_{DS}(x, \theta_i)$ from $g_{DS}(x, \theta_i + \frac{\pi}{2})$ which is the line integral at an aspect angle turned 90° . The problem becomes clear in the Fourier domain. By the line projection theorem $\epsilon(\theta, f_k)$ is the F.T. of $\sigma_0(\xi, n)$ restricted to region R_{DS} with aspect angle θ . $\epsilon(\theta + \frac{\pi}{2}, f_k)$ is the F.T. of $\sigma_0(\xi, n)$ restricted to region R_{DS} with aspect angle $\theta + \frac{\pi}{2}$. This is shown in Figure 6. Now it is obvious that $\epsilon(\theta, f_k)$ and $\epsilon(\theta + \frac{\pi}{2}, f_k)$ represent distinct spatial frequency components of the target in R_{DS} , except for the nonexisting DC term. Hence, $\epsilon(\theta, f_k)$ cannot be inferred from $\epsilon(\theta + \frac{\pi}{2}, f_k)$ which means that under the constraints in this model, double shadowing cannot be corrected.

Conclusion

The RAT-SCAT site data has been rederived with reflectivity change and shadowing effect considerations being taken into consideration. The reflectivity change brings about a relation between the target reflectivity and the data, which is not really a Fourier Transform relation. Consequently, solving it is a difficult analytic problem. The shadowing effect has been partially solved by combining data with aspect angles π radians apart. Further correction due to double shadowing is believed to be impossible under the constraints of this model.

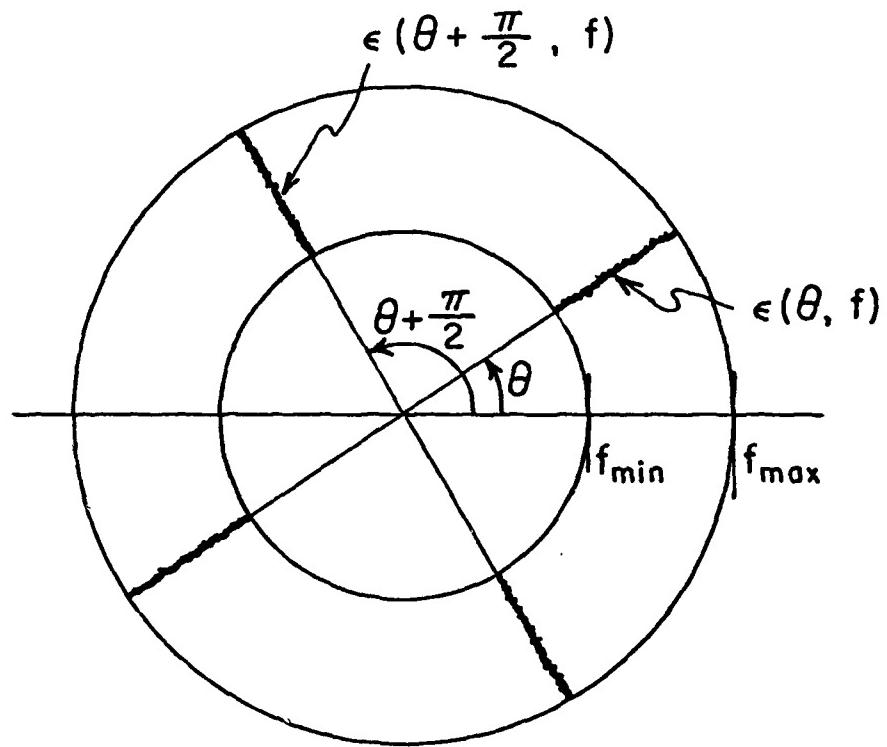


Fig. 6. F.T. $\epsilon(\theta, f)$ of the target reflectivity restricted to the double shadow region R_{DS} .

References

1. N.N. Bojarski, "Three-dimensional Electromagnetic Short Pulse Inverse Scattering," Syracuse University Research Corp., Syracuse, N.Y., SPL-67-3, Feb. 1967.
2. W.B. Brown and R.J. Fedorowicz, "Synthetic Radar Imaging of a Rotating Target," 13th Annual Radar Symp., Seattle, Wash., June 1967.
3. E.N. Leith, "Quasi-Holographic Techniques in the Microwave Region," Proc. of IEEE, Vol. 59, No. 9, September 1971.
4. R.N. Bracewell, "Strip Integration in Radio Astronomy," Australian J. Phys., Vol. 9, pp. 198.
5. J.L. Walker, "Range-Doppler Imaging of Rotating Object," Dissertation, University of Michigan, 1974.
6. J.F.A. Ormsby, N.M. Tomljanovich, H.S. Ostrowsky, M.R. Weiss, "Analytic Coherent Radar Techniques for Target Mappin," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-6, No.3, May 1970.
7. C.C. Chen and H.C. Andrews, "Turntable Radar Imaging," USCIP Report 800, March 31, 1978.
8. W.B. Brown and R.J. Fedorowicz, "Range-Doppler Imaging with Motion Through Resolution Cells," IEEE Trans. on Aerospace and Electronic Systems, Jan. 1969, pp. 98.
9. Kiyo Tomiyasu, "Tutorial Review of Synthetic-Aperture Radar (SAR) with Applications to Imaging of the Ocean

Surface," Proc. IEEE, Vol. 66, No. 5, May 1978, pp. 563.

10. J. Rheinstein, "Backscatter from Spheres: A Short Pulse View," Lincoln Laboratory Technical Report No. 414, April, 1966.

11. R.O. Harger, Synthetic Aperture Radar Systems, Academic Press, 1970.

Appendix A. Formulation of received data for RAT-SCAT system.

The Waveform transmitted from the radar R is assumed to be of the form $f(t) = \cos(2\pi f_k t + \phi)$ where f is the carrier frequency. Waves hitting target points lying at range $(r_0 - x)$ and reflected back towards R will have the form

$$f\left(t - \frac{2r_0 - 2x}{c}\right) \quad (1)$$

where $(2r_0 - 2x)/c$ is the path delay. Because of the specular reflectivity $\sigma(\xi, n; \theta_i)$ of the target, the signal reflection will be modulated by the following line projection integral $g_I(x, \theta_i)$ of points lying on the line of constant range $r_0 - x$,

$$g_I(x, \theta_i) = A \iint_{R_I} \sigma(\xi, n; \theta_i) \delta(\xi \cos \theta_i + n \sin \theta_i - x) d\xi dn \quad (2)$$

where A is a constant representing propagation attenuation and far field illumination pattern. Besides the target reflected signal, a reference signal is also received reflected from the reference sphere. This real signal (called)

in-phase component) and its $\frac{\pi}{2}$ radian delayed version (called quadrature component) together called the reference signal will be written in a compact complex form

$$r(t) = Be^{+j\left[2\pi f_k(t - \frac{2r_1}{c}) + \phi\right]} \quad (3)$$

The signal received from target reflection over 11 ranges is

$$S_I(i, k, t) = \iint_{R_I} Af\left(t - \frac{2r_0 - 2(\xi \cos \theta_i + \eta \sin \theta_i)}{c}\right) \sigma(\xi, \eta; \theta_i) d\xi d\eta$$

$$S_I(i, k, t) = \int_{-a}^a dx \iint_{R_I} Af\left(t - \frac{2r_0 - 2x}{c}\right) \sigma(\xi, \eta; \theta_i) \sigma(\xi \cos \theta_i + \eta \sin \theta_i - x) d\xi d\eta$$

Mixing $S_I(i, k, t)$ with (3) and integrating over time to get only the temporal D.C. term, we get

$$D_I(i, k) = \frac{1}{T} \int_0^T r(t) \cdot S_I(i, k, t) dt$$

where T is the dwell time of the signal at f_k .

$$\begin{aligned} &= \frac{1}{T} \int_0^T \iint_{R_I} AB e^{+j\left[2\pi f_k\left(t - \frac{2r_1}{c}\right) + \phi\right]} f\left(t - \frac{2r_0 - 2(\xi \cos \theta_i + \eta \sin \theta_i)}{c}\right) \sigma(\xi, \eta; \theta_i) d\xi d\eta \\ &= \int_{t=-\infty}^{\infty} dt \iint_{R_I} AB e^{+j\left[2\pi f_k\left(t - \frac{2r_1}{c}\right) + \phi\right]} \cos \left[2\pi f_k \left(t - \frac{2r_0 - 2(\xi \cos \theta_i + \eta \sin \theta_i)}{c}\right) + \phi\right] \sigma(\xi, \eta; \theta_i) d\xi d\eta \end{aligned}$$

Using the relation $\cos \alpha = \frac{1}{2}(e^{j\alpha} + e^{-j\alpha})$

$$D_I(i, k) = \iint_{R_I} \frac{1}{2} AB\sigma(\xi, n; \theta_i) \cdot$$

$$\left[\frac{1}{T} \int_0^T e^{j2\pi f_k [2t - \frac{2r_0 + 2r_1 - 2(\xi \cos \theta_i + n \sin \theta_i)}{c}]} e^{j2\phi} dt \right. \\ \left. + e^{j2\pi f_k [\frac{2r_0 - 2r_1 - 2(\xi \cos \theta_i + n \sin \theta_i)}{c}]} d\xi dn \right]$$

Note that the second term in the square brackets is the temporal DC term unaffected by the integration. Assuming that T is long enough so that the first term is averaged out we are left with

$$D_I(i, k) = e^{j2\pi (\frac{2f_k}{c}) (r_0 - r_1)} \iint_{R_I} \sigma(\xi, n; \theta_i) e^{j2\pi (\frac{2f_k}{c}) (\xi \cos \theta_i + n \sin \theta_i)} d\xi dn$$

Let

$$z_I(\theta_i, f_k) = \iint_{R_I} \sigma(\xi, n; \theta_i) e^{-j2\pi (\frac{2f_k}{c}) (\xi \cos \theta_i + n \sin \theta_i)} d\xi dn$$

Then the recorded data is in this form

$$D_I(i, k) = e^{j2\pi (\frac{2f_k}{c}) (r_0 - r_1)} z_I(\theta_i, f_k)$$

Earlier, we have used a complex reference signal to beat against the target reflected signal. As a result, we get a complex factor in the integrand of $z_I(\theta_i, f_k)$ which becomes the Fourier Transform of the target reflectivity if $\sigma(\xi, n; \theta_i) = \sigma_0(\xi, n)$.

3.5 Blind OTF Restoration

John B. Morton and Harry C. Andrews

The methods of Cannon and Cole for finding the magnitude of the OTF have been extended to obtaining the phase of the OTF as well. The progress of this research has been reported on in several preceding semiannual reports (USCIP 800,770,740). The technique utilizes the method of Knox to extend his results of finding the complete Fourier description of an object from its autocorrelation and other information. In our method, an image is partitioned into subsections which are then Fourier transformed, subsequently autocorrelated in the frequency domain, and then statistically averaged. Under certain conditions both magnitudes and phases converge for the complete OTF description. Results of this technique are then incorporated in a Wiener filter for completion of the restoration. The numerator of the filter is the complex conjugate of the computed OTF and the denominator is the computed power spectrum of the blurred image.

Real world (non simulated) in camera distortions are obtained by various physical acts of violence upon the camera during exposure. The computed MTF, phases for the OTF, blurred images and restored objects are presented below.

This section presents the results for three photographically induced blurs. To induce the first two blurs the camera was jiggled and vibrated during exposure. The third blurred image was obtained from a private source; the blur was apparently the result of an incorrect use of the camera.

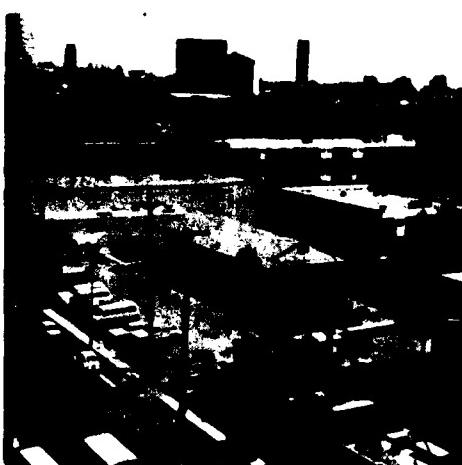
The blurred photographs were digitized to 512 x 512 pixels and the estimates of the OTF were made using 512 x 512 blurred images. A 50% overlapping of subimages gave 225 subimages of 64 x 64 pixels each. The estimate of the magnitude of the OTF was made via the method of Cannon. The phase estimate was obtained using a Parzen window and using recursion.

For the first two blurred images the restorations were made on each of the four 256 x 256 pixel quadrants of the 512 x 512 pixel blurred image. For the third blurred image four restorations of 256 x 256 pixels were made; the four restorations centered prominent features within the 512 x 512 pixel blurred images.

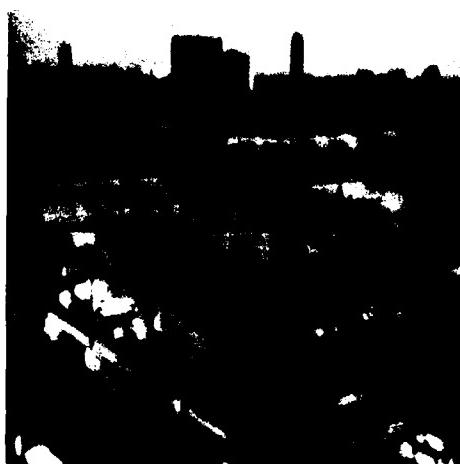
In Figure 1 is the same scene before and after the photographically induced blur. Note the "before" picture is earlier in time and is not used in the restoration. Figure 2 presents perspective plots of the estimates of the magnitude and phase of the OTF.

Figure 3-6 present the results of the restorations on the four 256 x 256 pixel quadrants of the 512 x 512 pixel blurred image. In each of figures 3-6 is presented the degraded quadrant together with a restoration using the estimate of the magnitude and phase of the OTF and a restoration using the estimate of the magnitude of the OTF and estimating the phase of the OTF to be zero.

Improvement is evident. In addition to an improvement in sharpness, some objects that were not recognizable in the blurred images are recognizable after restoration. In Figure 3a the vertical columns of the building in the northeast quadrant of the image are not resolved. In the restorations in Figure 3b and c the columns are resolved.



(a)



(b)

Figure 1 Scene before and after photographically induced blur

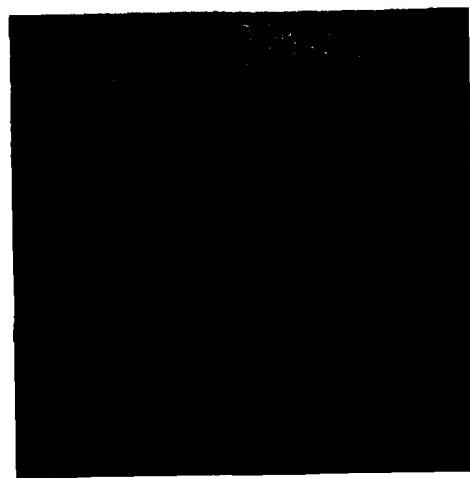
In Figure 4a the object on top of the tower is not recognizable. In the restorations in Figures 4b and c it is seen to be a ball. In addition, in Figures 4b and c note the improvement in definition of the windows and structure of the building which is in the center of the right-hand side of the frame.

In the restorations of Figures 5b and c the lines of the crosswalk are now defined. Additionally, there is better definition in the cars; it is now possible to recognize the Volkswagen as a Volkswagen. Note the increased resolution in the windows of the minibus.

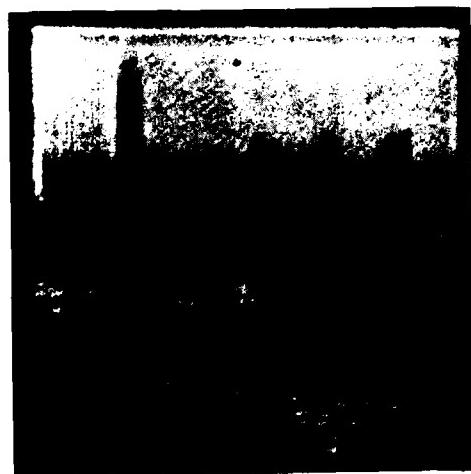
In Figures 7-12 are presented the results corresponding to the second photographically induced blur. Again, an improvement in sharpness and increased resolution is observed. In Figures 9b and c note the increased sharpness and definition in the tree in the south-west quadrant, the tree in the center of the right-hand side of the frames, and the trees along the top of the frames. Additionally, note the increased sharpness in the cars and building.

In Figures 10b and c there is better definition in the window panes and ledges. Note the increased sharpness in the trees of Figures 12b and c.

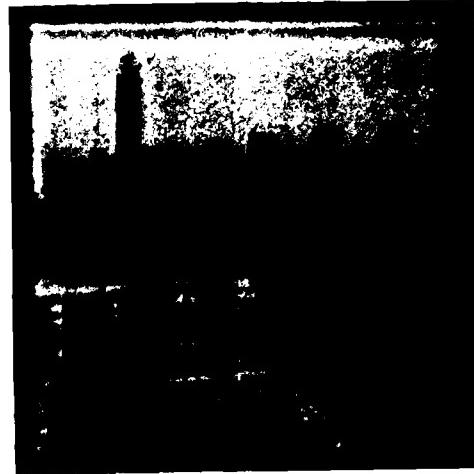
Figures 13-15 present the results for the third real world blurred image. Figures 13 and 14 present restorations of selected 256 x 256 pixel subimages of the larger 512 x 512 blurred image. Again, improvement is evident. For example, in Figure 13b note the increased sharpness of the edge content compared to Figure 13a. Additionally, in Figure 13b note the better definition in the rocks.



(a) degraded

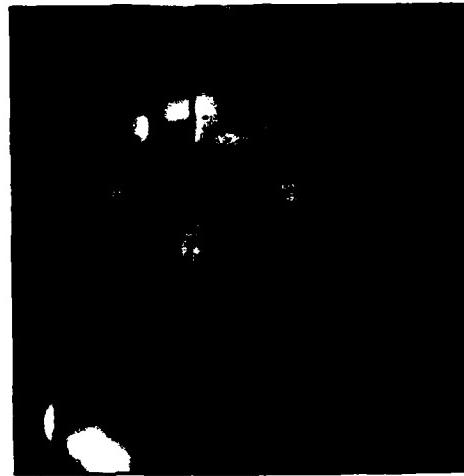


(b) phase of OTF estimated as 0



(c) phase estimate = Estimate 2

Figure 4 Blurred image and restorations



(a) degraded

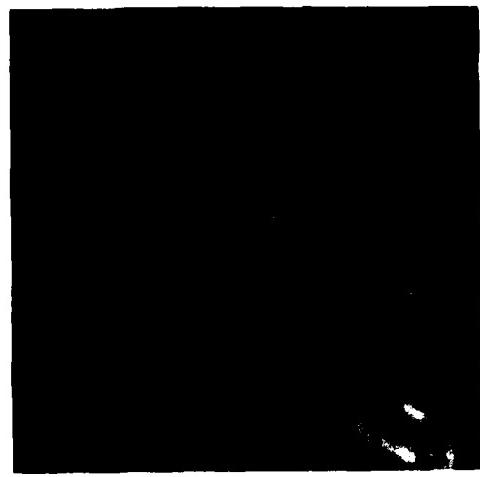


(b) phase of OTF estimated as 0



(c) phase estimate = Estimate 2

Figure 5 Blurred image and restorations



(a) degraded

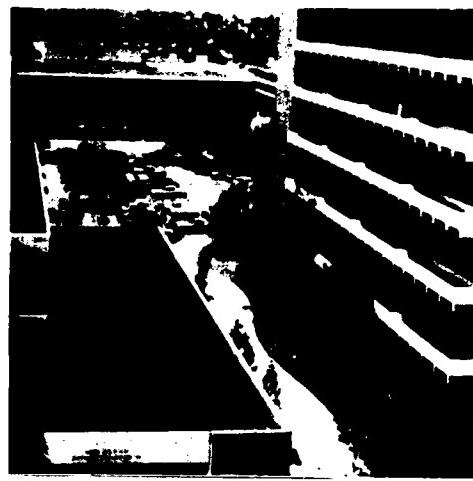


(b) phase of OTF estimated as 0



(c) phase estimate = Estimate 2

Figure 6 Blurred image and restorations

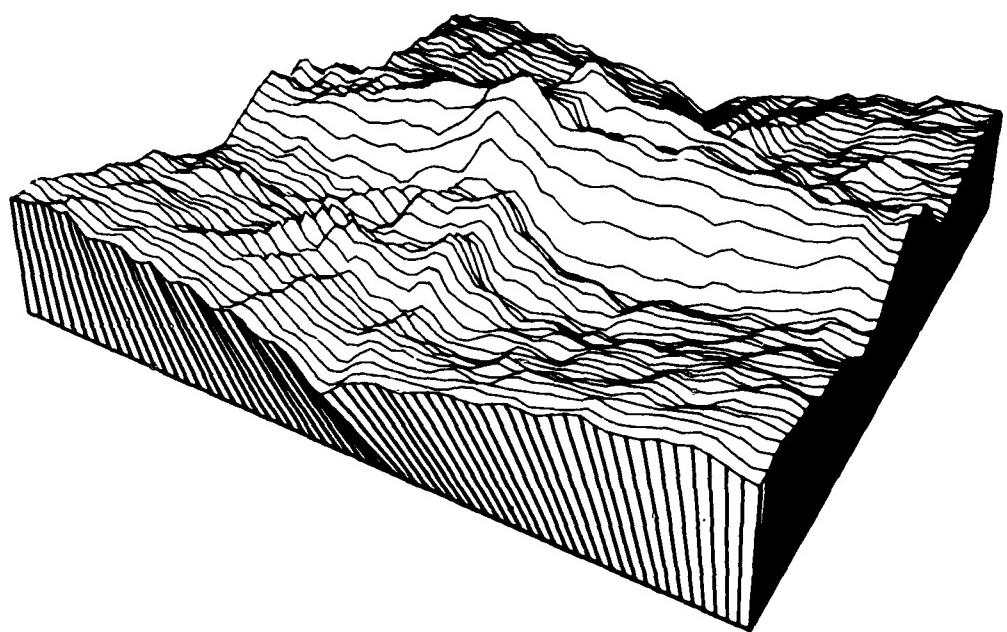


(a)

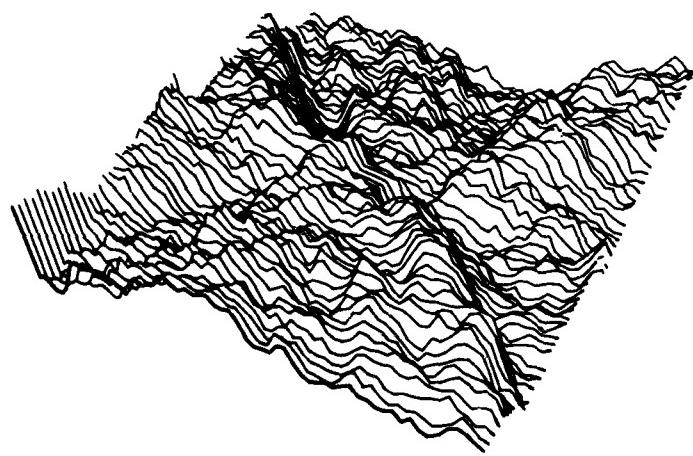


(b)

Figure 7 Scene before and after photographically induced blur

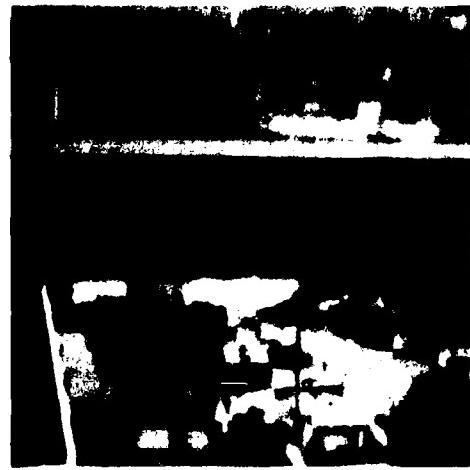


(a) $|\hat{H}|$

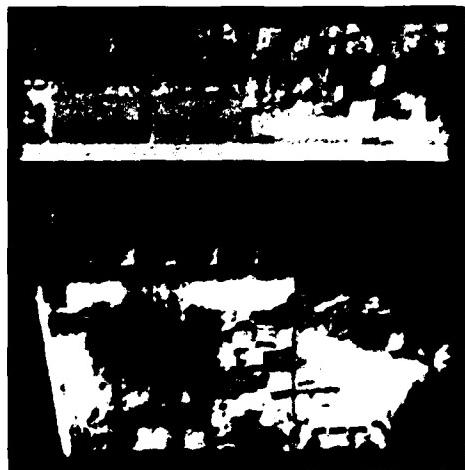


(b) $\hat{\theta}$ (minimum = -.69 radians,
maximum = .69 radians)

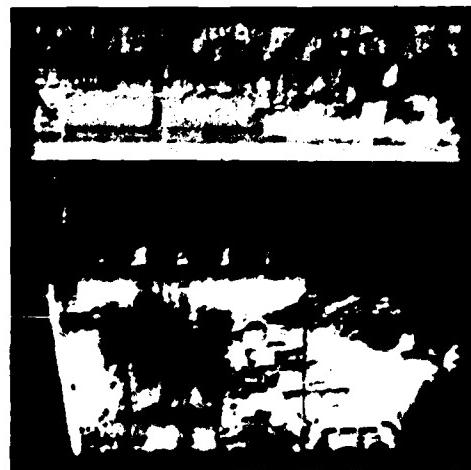
Figure 8 Estimates of magnitude and phase of OTF



(a) degraded

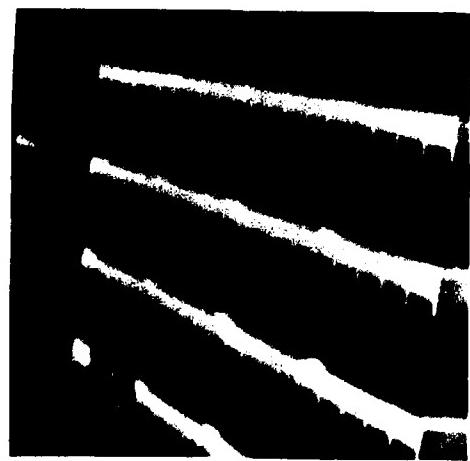


(b) phase of OTF estimated as 0

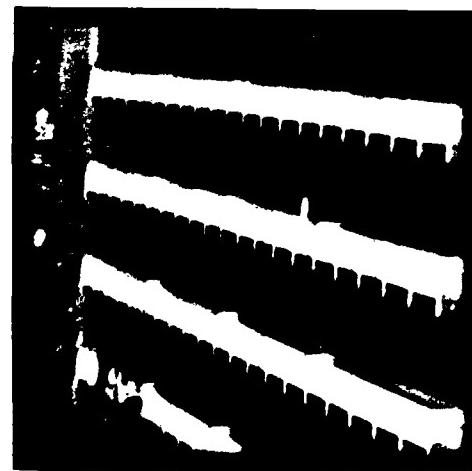


(c) phase estimate = Estimate 2

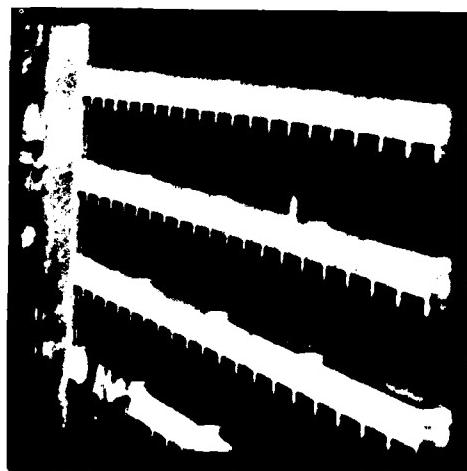
Figure 9 Blurred image and restorations



(a) degraded

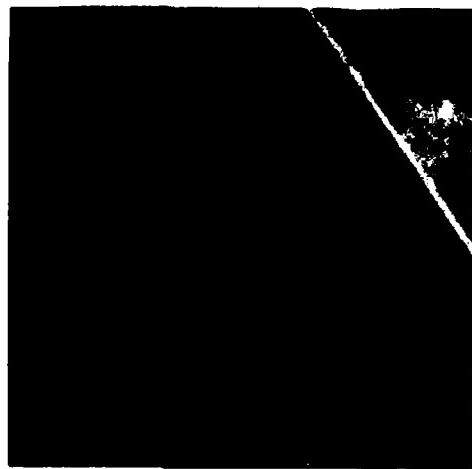


(b) phase of OTF estimated as 0



(c) phase estimate = Estimate 2

Figure 10 Blurred image and restorations



(a) degraded



(b) phase of OTF estimated as 0

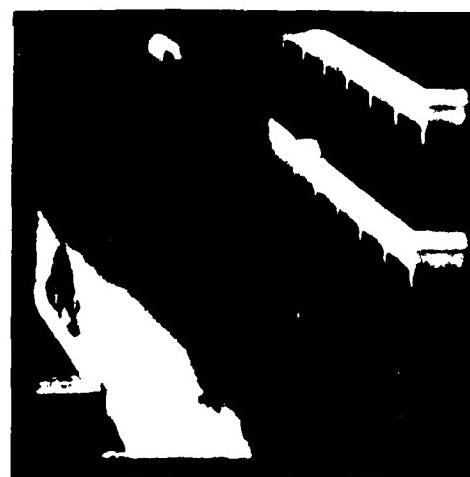


(c) phase estimate = Estimate 2

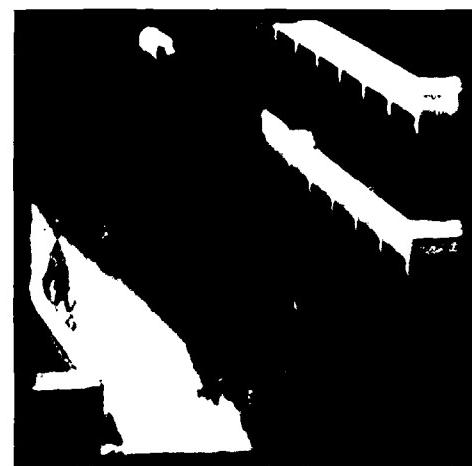
Figure 11 Blurred image and restorations



(a) degraded



(b) phase of OTF estimated as 0



(c) phase estimate = Estimate 2

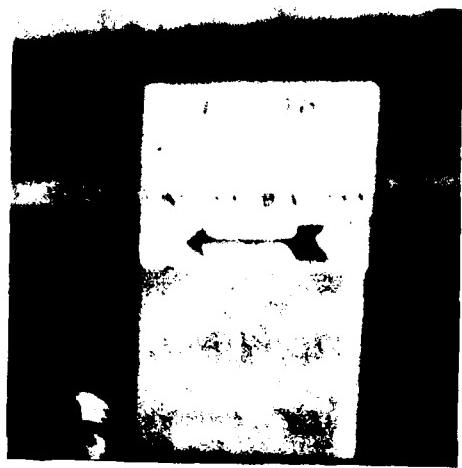
Figure 12 Blurred image and restorations



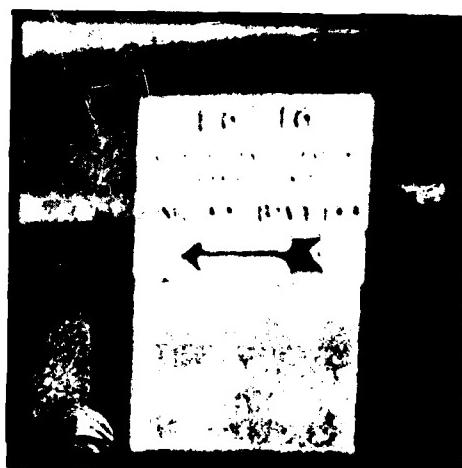
(a) degraded



(b) restored



(c) degraded



(d) restored

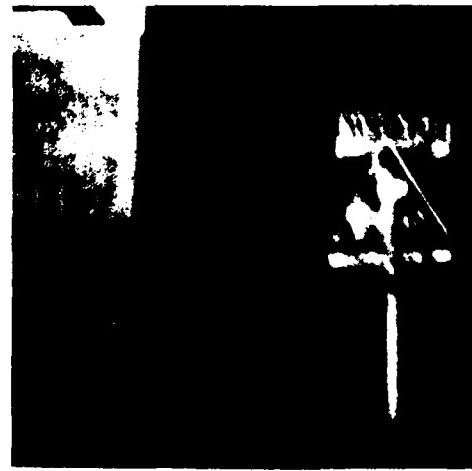
Figure 13 Blurred images and restorations



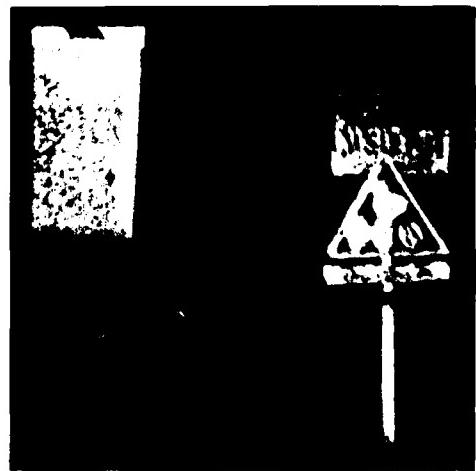
(a) degraded



(b) restored



(c) degraded



(d) restored

Figure 14 Blurred images and restorations

In Figure 1b there are what appears to be degraded point sources. For example, note what appears to be a degraded point source slightly below the center of the frame on the left-hand side of the frame. This affords one with the opportunity of estimating the magnitude and phase of the OTF using the degraded point source.

In Figure 16 is a comparison of the results using the techniques forming the basis of the research reported herein and results obtained using a degraded point source. Figures 16a and b are the same as Figures 5a and c, respectively, repeated here for convenience. Figure 16a is the degraded image. Figure 16b is the restoration obtained using the techniques reported herein.

3.6 Target Motion Induced Radar Imaging

Chung-Ching Chen and Harry C. Andrews

Imaging from ground-based (stationary) radars of moving targets is often possible by utilizing a "synthetic aperture" developed from the target motion itself. An aircraft is imaged from both a straight flight and a turn with recognizable results. Analysis shows that two phase components exist in the radar return, one being gross velocity induced, the other being interscatterer interference within the target itself. The former phase must be removed prior to imaging and techniques are developed for this task. Coherence processing intervals, range collapsing, and range re-alignment are all examined herein.

Introduction

In order to reconstruct a radar image of some target from its signal returns, two conditions have to be satisfied. First, the returned data has to have some kind of two-dimensional format. Second, the radar imaging geometry must be such that the return from each pulse or signature contains different (could be only "slight") information about the target. Degree of freedom analysis on the radar returns provides an attempt at evaluating the capabilities of the imaging system by examining the extent to which the above conditions are satisfied.

In the usual case of a pulsing radar, the return from a single pulse contains timing or range information, while the direction, called azimuth, along which the many pulse returns are aligned side by side, contains cross range information, and thus the first requirement for imaging is readily met. The second requirement demands that each pulse return be different. To accomplish this it is necessary to create a relative motion between the target and radar in such a way that the aspect angles of the target as observed from the radar are different for different pulses so that the cross range or azimuth information can be inferred. In this report, we look into a ground-based radar system in which a target aircraft is imaged by its own motion induced doppler.

Figure 1 shows the flight path of a target aircraft which has an overall length of approximately 80 feet and wing span of about 70 feet. Two portions of the flight path along which the data were obtained for imaging will be called interval 1 and interval 2, as shown in Fig. 1. The first interval is when the airplane was flying straight, at angles approximately 30 to 15 off broadside, whereby the second interval occurs when the airplane was making a standard left turn.

Figure 2 is a reproduction of the first interval of Fig. 1. The aspect angle of the target center viewed from the radar undergoes a change ϕ , which in this case is the same as the aspect angle of the target body with respect to the radar line of sight (LOS). In fact, it can be shown that it is the latter, and not the former angle change, which provides the azimuthal information. For this reason, we redraw Fig. 2 using the target center as the origin of the coordinate system. This becomes Fig. 3. Observe a close resemblance of Fig. 3 to the rotational geometry of a turntable system [1]. The flight path of the second interval is depicted in Fig. 4.

Preprocessing

For most practical purposes, the radar imaging system which determines the relation between the data returns and the reflectivities of the target can be considered linear [2] and the system classification method developed elsewhere can be used to decide the ways to reconstruct the reflectivities directly from the raw data. This situation is depicted in Fig. 5. In other words, the data return $g(x,y)$ is a linear transformation of target reflectivity function $f(\xi,n)$ through the radar signal radiation and the echo reception. For ease of presentation we will assume that both g and f in Fig. 5 are discrete so that the system can be represented by a matrix $[H]$ and g and f are vectors [3]. Depending on the waveforms of transmitted signals, (e.g. short pulse, linear FM pulse, or step-frequency waveforms) and the imaging geometries (e.g., shape and size of target, direction of relative motion, resolution required, etc.), the radar imaging systems represent a wide spectrum of the classes. Once the relation $[H]$ between the reflectivity and data is (precisely) decided by the flight or radar data, a straightforward reconstruction of f and g

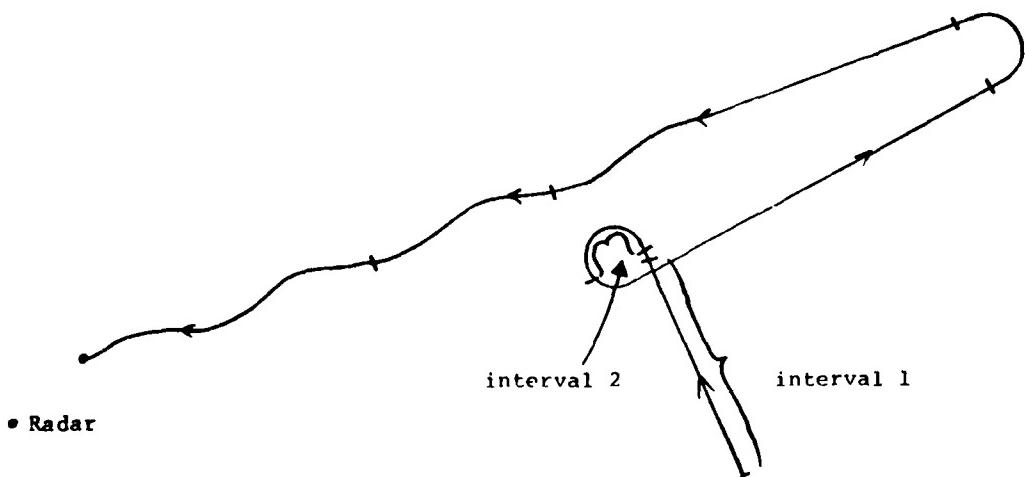


Fig. 1. Overall flight path of target data.

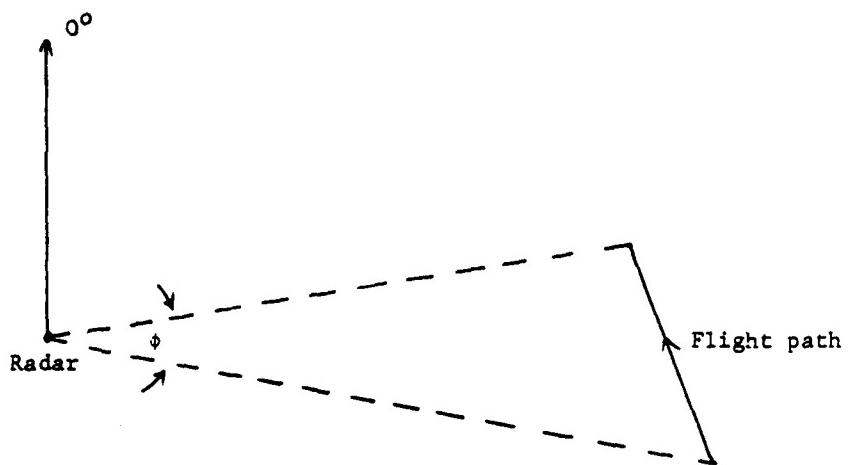


Fig. 2. Flight geometry of first interval

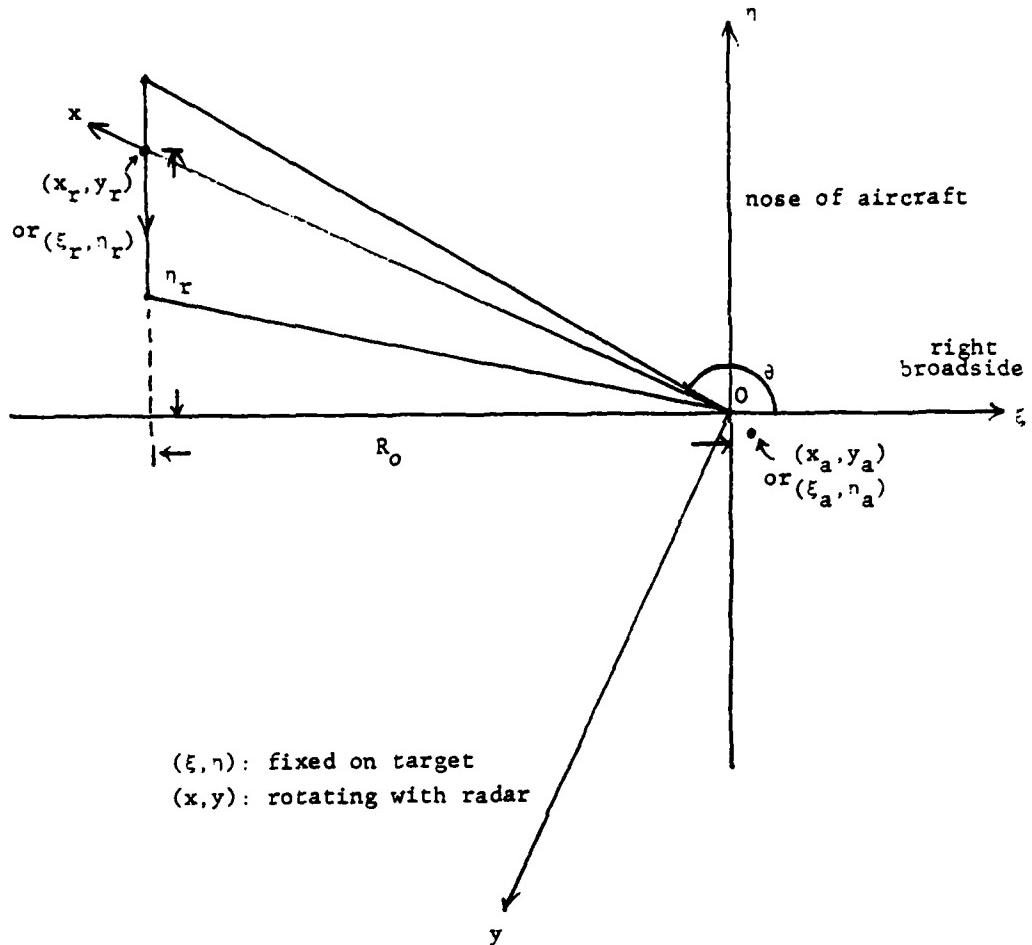


Fig. 3. Trajectory of radar in first interval with respect to the target center.

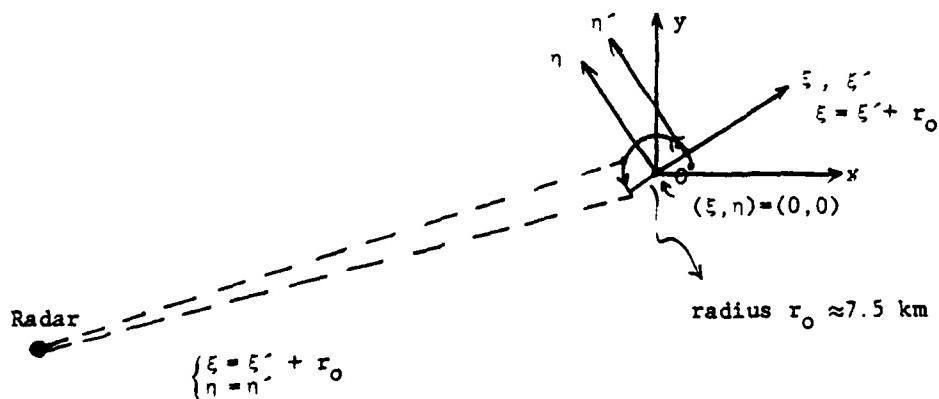


Fig. 4. Geometry of the second interval of the flight.

can be achieved by applying the pseudoinverse of $[H]$ to g yielding a minimum square error reconstruction.

The above reconstruction scheme, although straightforward in theory, usually involves a great deal of computation because of the complexity of $[H]$. In the worse case, one would expect to resort to a full singular value decomposition (SVD) to find $[H]^{-1}$. Certainly a decomposition of $[H]$ such that the structure of the imaging geometry can be better utilized would warrant the efforts in many cases.

A perceivable way to accomplish this is to do some preprocessing upon the raw data such that the resultant data have a much simplified relation to the reflectivity than the raw data itself. Diagrammatically, $[H]$ can be replaced by a cascaded system of $[H_1]$ and $[H_2]$ as in Fig. 6 and f can be estimated by multiplying $[H_2]^{-1}$, followed by $[H_1]^{-1}$, to g with the hope that $[H_1]$ would be so simplified in structure or so small in size compared to $[H]$ that the extra effort on $[H_2]^{-1}$ would be warranted. For this purpose $[H_2]^{-1}$ is called preprocessing. Examples of preprocessing are: range alignment, presumming, de-chirping, and motion compensation. Some of them will be discussed in the following sections.

Range Curvature and Range Bin Alignment

In general, the radar return of the signal pulse from the target provides the range information while the history of the returns along some range bin provide azimuthal information. These two sources of information could be coupled such that a separable or even separate processing would not be adequate to recover the information to the extent of accuracy one pursues. There are two major sources of non-separability in the radar system: range walking and

data misalignment. We now describe the phenomena and propose methods to avoid or correct them.

A. Range Curvature

A single radar pulse return contains the information about the surfaces or lines whose points are equi-distant from the radar transmitter. These surfaces or lines can be resolved by the timing (for short pulse) or range compression (for long duration linear FM-like pulse) techniques. Since the range direction has been compressed and resolved in our source data, the simplest way to resolve the azimuth would be to do one-dimensional processing along cross range direction. This requires that each particular point have contribution to only those range bins which are aligned for azimuthal processing. Such is the case for low or medium resolution SAR imaging with aligned returns. As the resolution requirement becomes greater and greater recently, one is usually forced to reduce the range bin width and/or to increase the azimuthal interval over which the data are to be processed coherently. Both of these would eventually create range curvature problems since the surfaces of constant range as mapped on the target move further away as the relative motion between the radar and the target continues.

Since range curvature depends on the range bin width, ideally one can avoid range curvature by increasing the range bin width. This means sacrificing range resolution for the azimuthal resolution in the case of separable processing. It is not true, though, that the range curvature limits the width of coherent processing available. In fact, in the range curvature situation one can do some limited compensation by the techniques described in [4,5] or even full compensation by resorting to a non-separable model

for the imaging system [4] and relying on singular value decomposition (SVD) techniques. However, in all of our experiments we will always assume separable processing for ease of computation and implementation.

B. Range Alignment

In addition to the range curvature, there is another problem which hinders the separability of the processing: range misalignment. As described before, azimuthal processing operates upon the returns which came from target points at equal range. Thus precise timing or other schemes on returns of individual pulses to insure correct range bin alignment is of ultimate importance to warrant separable processing.

In the data of our radar system, range tracking is provided by a Poly/ Kalman estimator which tries to lock the first strong peak of each pulse return onto a specific range bin. For example, if the point on the target closest to the radar is the wing tip, then the wing tip returns of different pulses hopefully will be locked in the same range bins. Because of scintillation of the reflectivities, this range locking method is not always reliable and misalignment occurs from time to time.

Let $f_{t_1}(r)$ and $f_{t_2}(r)$ be recorded complex return (or our source data) from adjacent pulses where $t_2-t_1 = \Delta t$ is the pulse repetition interval (PRI) and r is the range assumed within one PRI. Because of the tiny aspect angle change in one PRI, if we consider only the magnitude of the returns, then

$$m_{t_1}(r + \Delta r) \approx m_{t_2}(r) , \text{ where } m_{t_1}(r) \triangleq |f_{t_1}(r)|$$

for some Δr , the amount of misalignment which we would like

to estimate. Define a correlation function between the two waveforms $m_{t_1}(r)$ and $m_{t_2}(r)$:

$$R(s) \triangleq \frac{\int_{-\infty}^{\infty} m_{t_1}(r) m_{t_2}(r-s) dr}{\left[\int_{-\infty}^{\infty} m_{t_1}^2(r) dr \int_{-\infty}^{\infty} m_{t_2}^2(r) dr \right]^{\frac{1}{2}}} \quad (1)$$

Then because $m_{t_1}(r+\Delta r) = m_{t_2}(r)$, from the Schwartz inequality we have that $R(s)$ will be maximal at $s = \Delta r$ and the amount of misalignment can thus be determined. It is observed from Eq. (1) that since the denominator is independent of $R(s)$, it can be dropped without affecting the peak location. Thus we could use

$$R'(s) \triangleq \int_{-\infty}^{\infty} m_{t_1}(r) m_{t_2}(r-s) dr$$

which is a straight convolution relation.

Motion Compensation

As described earlier, there are two kinds of phase variations induced by motion of the target: motion of the target center relative to the radar and that of the different target points relative to the target center as viewed from the radar. Only the latter contributes to the imaging ability of the radar. It can also be shown that the relation between the latter phase variation and the target reflectivity is a simple Fourier transformation in the azimuthal direction. Thus, a motion compensation of $[H_2]^{-1}$ which removes the effect of the motion of the target center

is highly desirable.

Two schemes of the motion compensation are proposed for our imaging.

First scheme: The flight path of the target center can be inferred from the timing of the pulse returns. For example, in the first interval after the flight path has been decided to be a straight line and the azimuth angle has been determined, the coefficients of the quadratic and other higher-order phases can be determined to remove the flight path phases and leave only the phase information relevant to the imaging.

Second scheme: Since the trajectory of a single target point is very similar to that of the target center, the returns from that point, if available, can as well be used as a reference to compensate for the target center motion. In fact, this is equivalent to considering this target point as the rotation center of the target. The phases of this reference point, as a function of azimuthal signatures, can then be subtracted from those of all the range bins at the corresponding signatures. Care should be exercised to assure two things: first, the size of the reference point must be small enough. This is because the size of the reference point decides the best possible azimuthal resolution. Second, for each signature, the reference range bin must correspond to the reference point if the advantage of a fast separable processing is to be taken. This requires range alignment as described before.

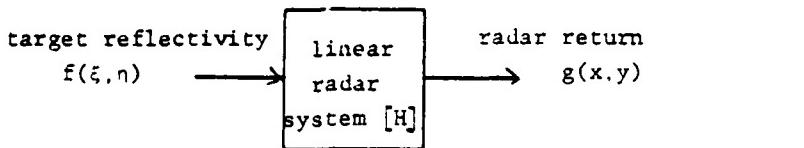
Presumming

The purpose of presumming is to remove the factor of oversampling in the azimuthal direction. Usually the radar

imaging system is oversampled in the azimuth direction because of a too high PRF. In the case of terrain imaging, oversampling is sometimes a result of not processing the whole antenna illumination pattern along the azimuth direction. In that case, the pattern width utilized or coherently processed determines the resolution of the image. In the case of aircraft imaging, the situation is different. Here the azimuthal width of the aircraft is so small that we would always like to make full use of the maximum width of the effective radar illumination pattern, which is the azimuthal length of the aircraft itself. Under this condition the PRF required is decided by the azimuth dimension on the aircraft and the azimuth resolution is decided by the signatures coherently processed. Thus, assuming other parameters fixed, a larger aircraft would require a higher minimum PRF to insure that no aliasing will occur in the final images. Also, since the effective antenna illumination (i.e., overall aircraft azimuthal length) is independent of the wavelength, λ , the minimum PRF or the resolution in the aircraft-imaging case would be functions of λ . This is in contrast to the ground terrain imaging cases where the full antenna illumination pattern width, which is proportional to λ , is to be fully used so that the resultant resolution is independent of the because of a cancelling effect. [1,2]

Let f_0 be the carrier frequency and L the length of the aircraft along the direction normal to LOS, as shown in Fig. 7. Let $\Delta\theta$ be the orientation change of the target as observed from the radar between two adjacent pulses, then the azimuthal frequency change will be

$$\Delta f_z \approx f_0 \Delta\theta$$



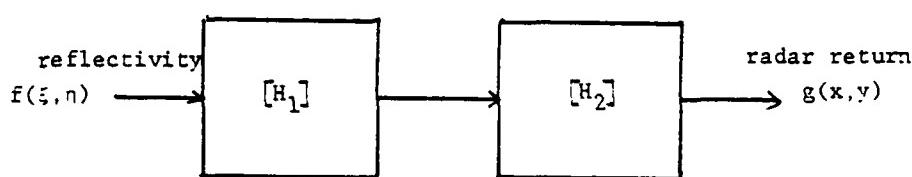
$$g = [H] f$$

$[H]$: Point spread function matrix (PSF)

$[H][H]^T$: correlation matrix

- determines systems degrees of freedom (DOF).

Fig. 5. Linear radar imaging system.



$$[H] = [H_2] [H_1]$$

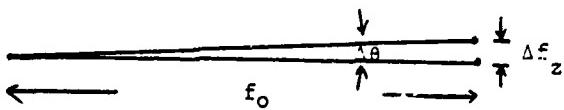
$$[H^{-1}] = [H_1]^{-1} [H_2]^{-1}$$

$[H_2]^{-1}$: Preprocessing

- Range alignment
- Range Walking
- Motion Compensation
- Presuming

$[H_1]^{-1}$: Ideally a Fourier transform

Fig. 6. Decomposition of $[H]$.



$$\ell = \frac{c}{2\Delta f_z} / L \quad \text{with } L: \text{azimuth length of target}$$

Fig. 7. Determination of oversampling factor.

This means a sampling interval of Δf_z in the azimuthal frequency domain, which in turn implies a non-ambiguous azimuthal time interval of $\frac{1}{\Delta f_z}$, or spatial interval of $\frac{c}{2\Delta f_z}$, in accordance with the sampling theorem.

The oversampling factor α can be determined by

$$\alpha = \frac{c}{2\Delta f_z} / L$$

To remove the oversampling to leave minimum useful data, a coherent low pass filter followed by sampling at a corresponding low rate should be applied. The effect of low-pass filtering is to remove the high frequency noise which otherwise would appear in the resultant image.

Experimental Results - First Interval

The mode of the radar system in which our source data was acquired was a wide band high range resolution mode. The transmitted pulse was a linear FM and the pulse returns have been compressed using matched filtering techniques in the radar receiver.

A condensed overall view of magnitude part of the first interval data is shown in Fig. 8 in which each row corresponds to the logarithm of the magnitude of the return from a single pulse. Only every 16th signature is shown in this figure. Recalling that this interval represents the radar returns when the target aircraft was flying toward a broadside position (Fig. 1), we presume that the first high-intensity bins correspond to the left wing tip and the next distinct strong returns are from the fuselage and nose. Note that the radar is to the left of this figure.

Then it can be perceived from Fig. 8 that the fuselage is at a greater and greater distance away from the wing tip along the range direction, as a result of closing-to-broadside during flight. It is also observed that while most portions of Fig. 8 seem pretty well range-aligned, other portions do need re-alignment before a separable processing can be implemented.

To present the data in detail all of the first 512 signatures are displayed in Fig. 9. The phase image (Fig. 9b) indicates clearly that the target points probably lie in range bin number 50 to 200, where a strong structure of phase relationships appear as a result of the coherent radar pulsing. This is also shown in the log magnitude picture Fig. 9a, although with less clarity. There is a transient region where the strength of the returns decreases gradually with the range or time. This phenomena is conjectured to be a result of multiple reflections on the target which took more time before re-radiating to the radar receiver.

To investigate further the behavior of the returns, only the regions of strong signal returns are kept and a sequence of 8192 signatures is shown in Fig. 10 with both log magnitude and the corresponding phase. Observe the quadratic-like phases along the flight direction due to the flight geometry, as analyzed earlier in this report.

Since the radar receiver has range compressed the signal returns we will need only to perform some azimuthal processing. For convenience we transpose the data so that the horizontal direction now denotes the signature or azimuth direction. Figure 11 shows the log magnitude and phase of typical signatures (signatures 8001 to 8512). To remove the quadratic phases from Fig. 11b, three options

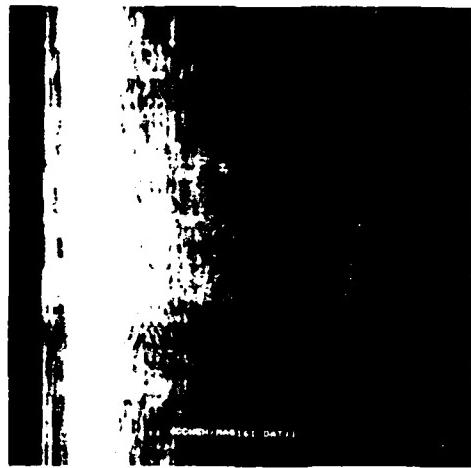
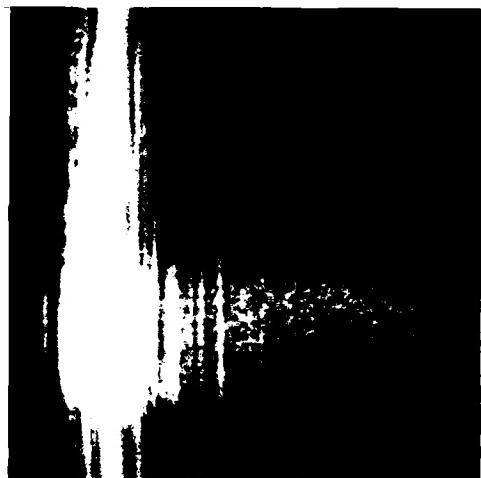
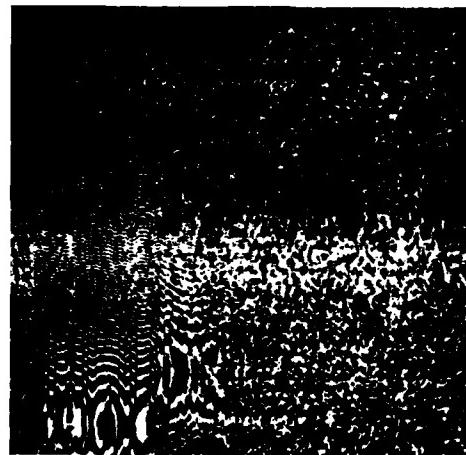


Fig. 8. Overall view of first interval data; log magnitude of every 16th pulse return.



(a) Log magnitude

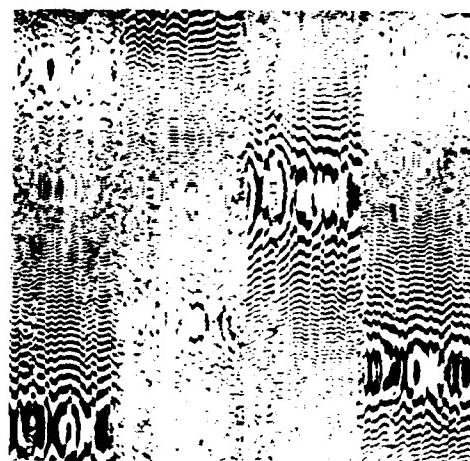


(b) Phase

Fig. 9. First 512 signatures of first interval data.



(a) Log magnitude

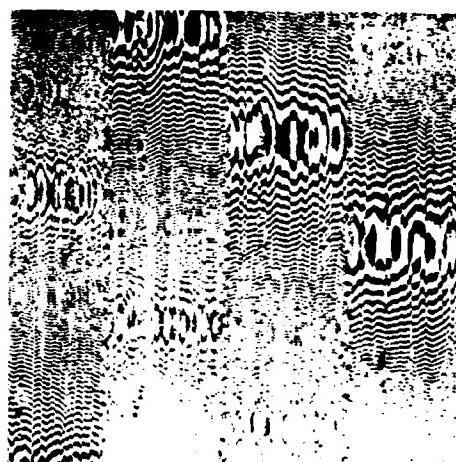


(b) Phase

signature number 1-2048



(c) Log magnitude



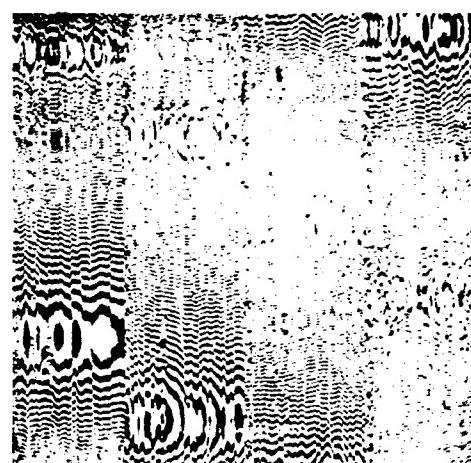
(d) Phase

signature number 2049-4096

Fig. 10. First interval data with 128 range bins stacked side by side.



(e) Log magnitude

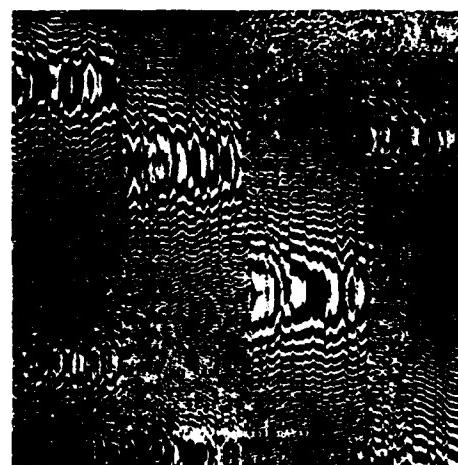


(f) Phase

signature number 4097-6144



(g) Log magnitude



(h) Phase

signature number 6145-8192

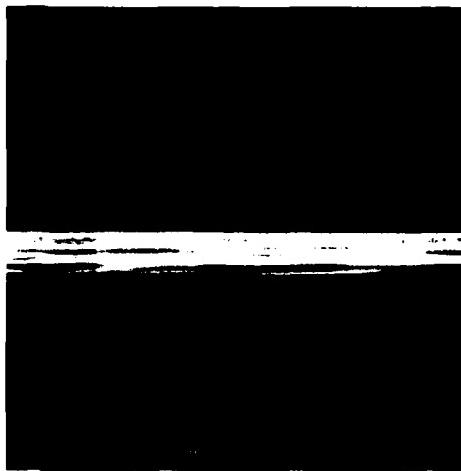
Fig. 10. (continued)

exist, the first two being similar:

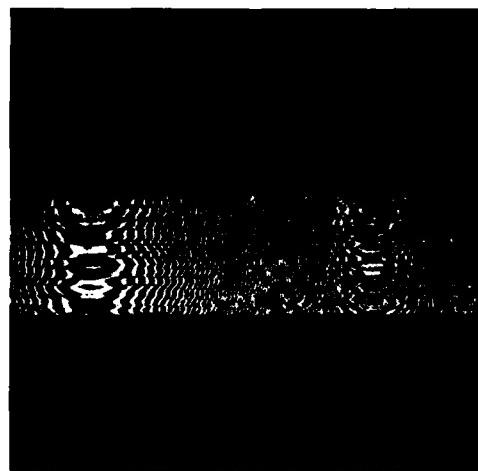
(A) Linear fitting the phase difference and subsequent integration: since the differentiation of quadratic phases is linear phases, a linear fit to the phase differences can be applied to determine the quadratic phase curvature. Figure 12a shows the azimuthal phase difference of Fig. 11b. Note that except for the phase wrap-around in the right half portion, the intensity, which is used to linearly encode the phase between $-\pi$ to π , looks quite linear. However, before a successful linear fit can be obtained, the phase-wrap problem has to be solved and this is usually not a very easy task. In fact, it is the rapid phase modulus phenomena in Fig. 11b that causes unwrapping Fig. 11b extremely difficult. We use a simple-minded scheme and the unwrapping of the phase difference, although not perfect, of Fig. 12a is shown in Fig. 12b, from which the linear portion of phase variation was estimated and removed to leave Fig. 12c. Since Fig 12c is still in the differentiation domain, an integration brings it back to the azimuthal phase domain, as depicted in Fig. 12d.

(B) Linear fitting the phase difference and quadratic subtraction: an alternative to applying the estimated linear-phase-difference is to subtract the estimated quadratic phase (from integration of estimated linear phase difference) from Fig. 11b directly. The result is shown in Fig. 12e. Magnitudes of azimuthal Fourier transforms of Figs. 12d and 12e are shown in Figs. 13a and 13b, which are very similar visually.

Since the azimuthal sampling rate has been determined to be greater than a factor of 50, the data after quadratic phase compensation can be reduced by a factor of 32 before Fourier transformations are applied. The result is shown in



(a) Log magnitude

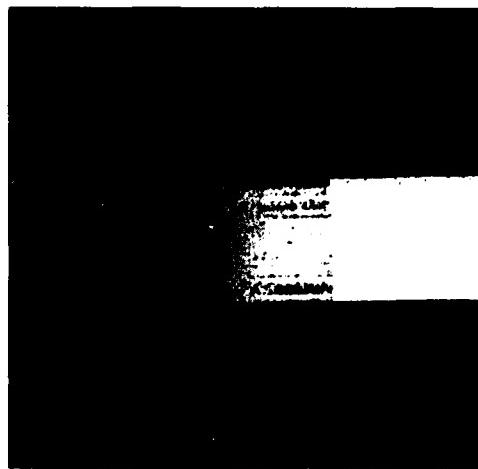


(b) Phase

Fig. 11. Signature number 8001-8512
(range direction is vertical).

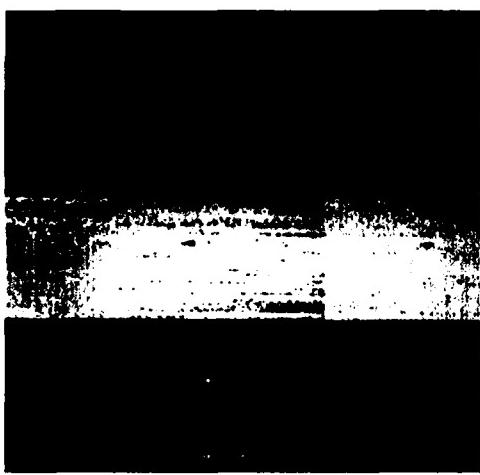


(a) Phase difference of
Fig. 11b

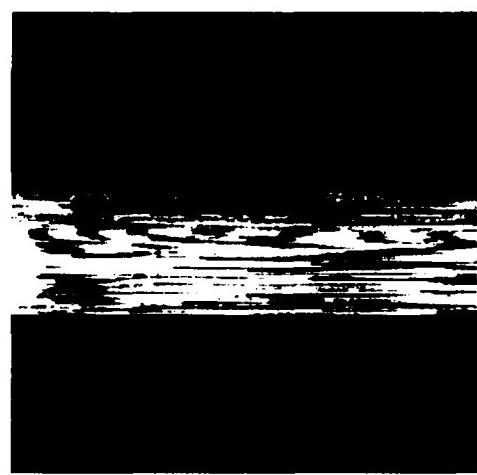


(b) Unwrapped version
of (a)

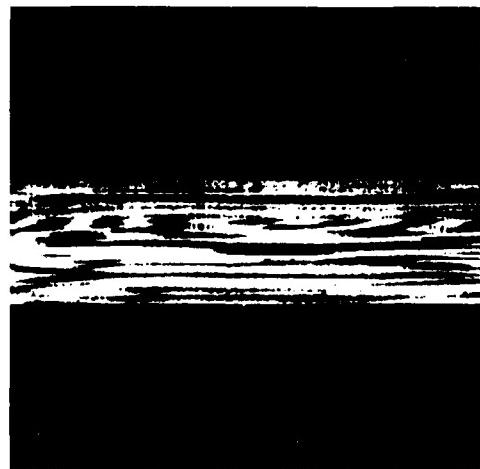
Fig. 12. Motion compensation on Fig. 11b using scheme of linear curve fitting.



(c) Linear phase removed
from (b)



(d) Phase integration
of (c)



(e) Quadratic phase removed
directly from (11b)

Fig. 12. (continued).

Fig. 13c. A comparison of Figs. 13a and 13c confirms the validity of the coherent presumming. Note that in Figs. 13a, b, and c the azimuthal bin width is much wider than the range bin width and a subsequent interpolation has to be done to properly scale the images.

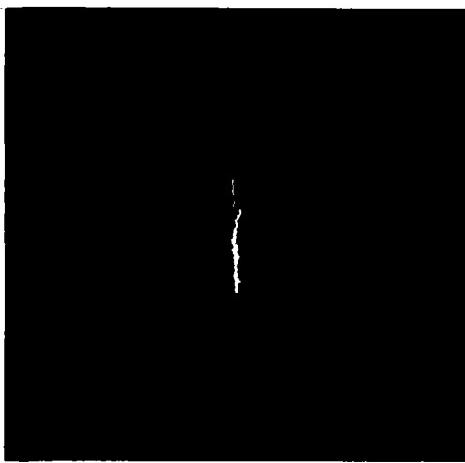
(C) Target point referencing: the above two schemes of removing phase variation due to target center motion are based on an assumption that the flight path is relatively straight during the coherence time. In other cases where the range trajectory is much more complicated than a low-order polynomial curve, the above schemes are expected to be more difficult. Another motion compensation scheme somewhat independent of the flight geometry and very simple in implementation is to use the signal returns from a reference point to estimate the history of the flight range trajectory. This single point can be thought of as the center of rotation of the target and its phases can be subtracted from those of all range bins to leave only the phase histories of all target points relative to this reference point. This was, in fact, the technique used in subsequent imaging.

Figure 14 is a series of processed aircraft images using the above reference point scheme. Consecutive pictures represent abutting 2048 signatures or 20-second flight time each. The images are linearly interpolated in azimuth to give the same range and azimuthal bin width such that the images are correctly scaled. Visually Fig. 14d is the best probably due to best range alignment of the data in that time interval.

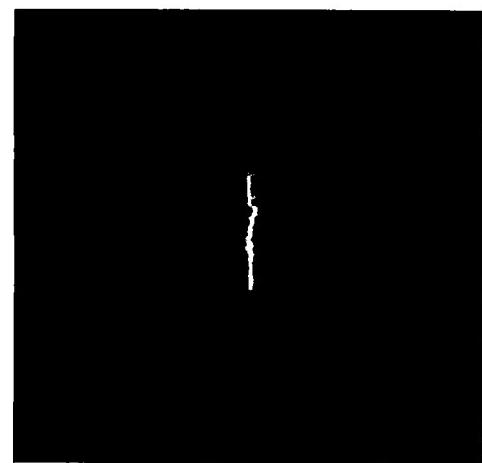
Ideally, an increase in coherence time should be accompanied with an equally increased amount of resolution. This is not the case in Fig. 15, where coherence times of 40

and 80 seconds are assumed. The conjecture is that the range curvature and range misalignment which tend to blur the images outplay the coherence time increase. As described in the previous section, one way to alleviate the range curvature problem is to use larger range bin widths. To test this, we used the same parameters as in Fig. 15b except the data in range dimension were reduced by a factor of two by coherent collapsing. The result shown in Fig. 16 is to be compared with Fig. 15b.

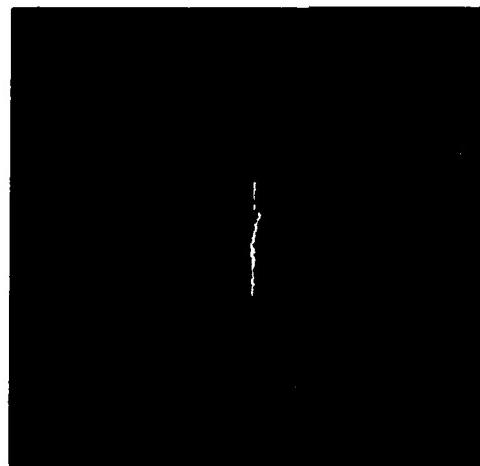
The phase variations of target points induced by the target motion is the key to the coherent radar imaging. As one can see from Fig. 9, the magnitudes of the radar returns which provide only range information are very similar from pulse to pulse and represent a great deal of redundancy. From the DOF point of view one would like to have approximately equal amounts of input and output data. Hence, it is conjectured that the phase portion of data alone is sufficient to give an image of a comparable quality. This would achieve a factor 2:1 in data reduction. Experimental result shown in Fig. 17 seems to support this conjecture. Intuitively speaking, the range bins where there are no strongly reflective target points have a random-like phase and are likely to spread their energy over the spectrum after the Fourier transform is taken in the azimuthal direction. On the other hand, target points of strong reflectivities give highly correlated azimuthal radar returns, resulting in clusterings of energy in the frequency domain corresponding to different azimuthal target points. In this way, the magnitudes of the returns do not play an important role in determining in which range bins lie the strong target points.



(a) Magnitude of Fourier
transform of 12d



(b) Magnitude of Fourier
transform of 12e



(c) Fourier transform of
collapsed version of (b)

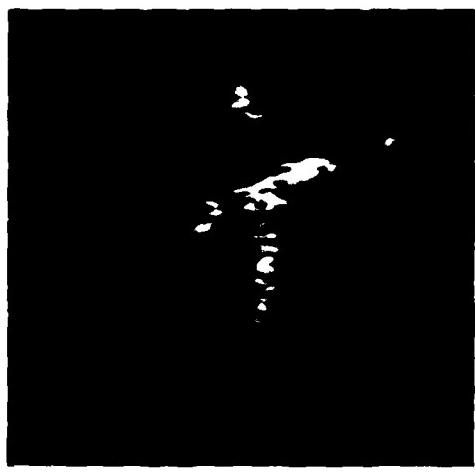
Fig. 13. Processing the phase data compensated by
schemes (A) and (B).



(a) 1st 20 seconds or 2048
signatures ($\sim 2.5^\circ$ aspect
change).



(b) 2nd 20 seconds

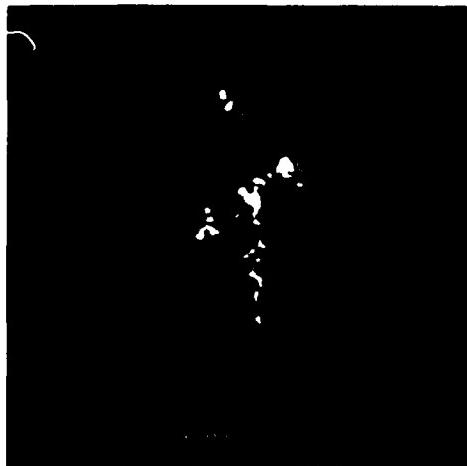


(c) 3rd 20 seconds



(d) 4th 20 seconds

Fig. 14. Aircraft radar images with abutting 20 second coherence time.



(a) 1st 40 seconds
($\approx 5^\circ$ aspect change)



(b) 2nd 40 seconds



(c) 1st 80 seconds
($\approx 10^\circ$)

Fig. 15. Aircraft images with different coherence times.



Fig. 16. 2:1 range collapsed
coherence time = 40
seconds.



Fig. 17. Same portion as Fig. 14d
phase information only.

Experimental Results - Second Interval

The first 8000 signatures of the second interval source data which were taken when the airplane was making a standard left turn are shown in Fig. 18 and Fig. 19. Unlike the straight flight, the phase plot here has a changing azimuthal structure due to the turning motion of the target, which creates complicated range and Doppler histories. In addition, there are several occasions when the range bins are seriously out of alignment. The overall view of Fig. 18 shows the changes of relative positions of nose, fuselage and wing tip due to the turn. A portion of data was taken when the airplane was nose into the radar and a series of resultant images are shown in Fig. 20 using the reference-point technique as a phase compensator. In this case the nose tip serves as a very good reference point as shown by the degree of sharpness of the nose in these images. Figure 21 shows images of different coherence times. Note that in Fig. 21b the shape of the fuselage has been clearly imaged. A coherence interval of 18° rotation of the target seems too much to give a satisfactory image as a result of overwhelming range walking.

The spread patterns close to the nose are due to the aircraft radar which was constantly scanning during the flight, presenting an object of changing reflectivity and violating the assumption that the target was a rigid body in the processing technique.

Range Re-Alignment Results

As is evident from Figs. 18 and 19 the radar breaks range lock quite often during the turn of the target aircraft. This is to be expected as different scatterers from the aircraft dominate the leading return of the radar reflection. Naturally when the radar breaks lock, one would not expect to be able to image without re-alignment

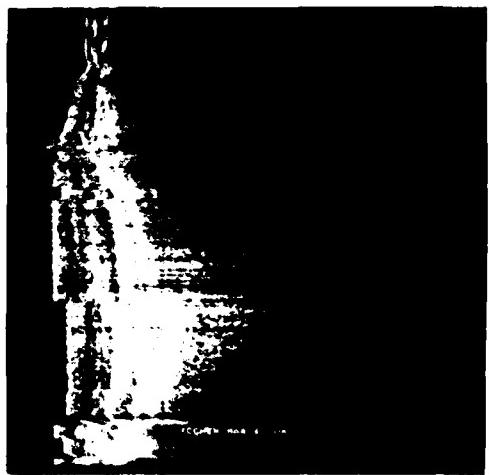
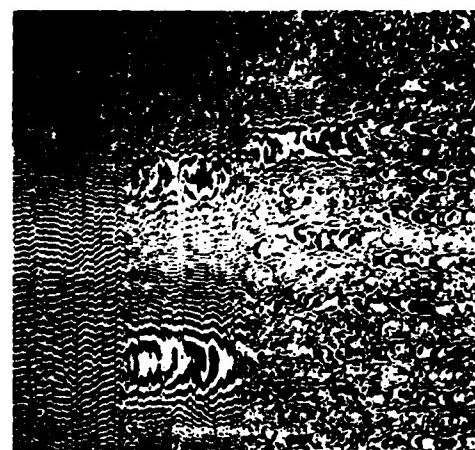


Fig. 18. Overall view of second interval data;
log magnitude of every 16th pulse return.

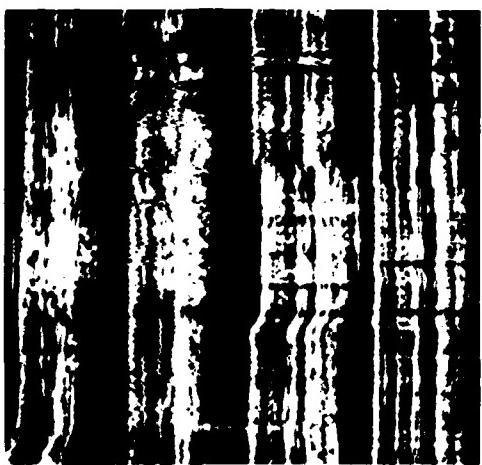


(a) Log magnitude

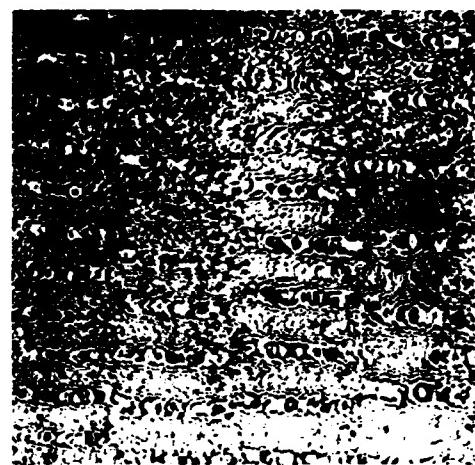


(b) Phase

signature number 1-2048



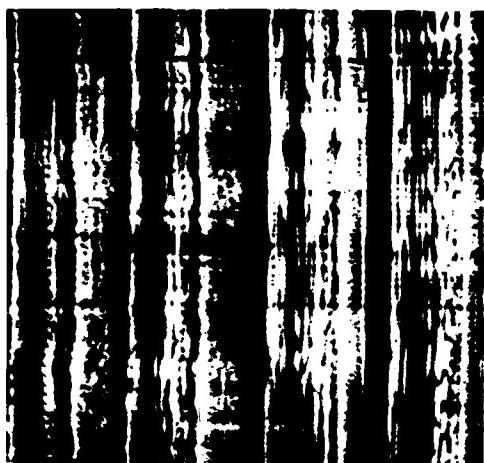
(c) Log magnitude



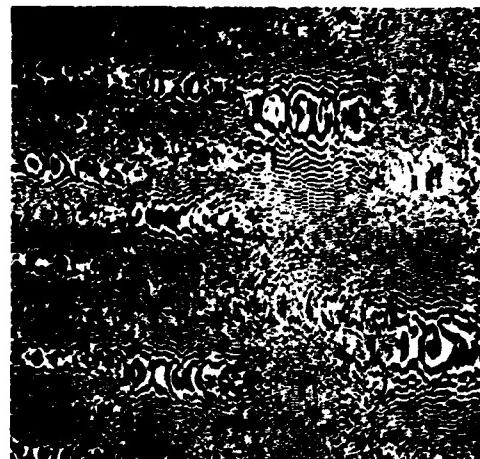
(d) Phase

signature number 2049-4096

Fig. 19. Second interval data with 128 range bins side by side.

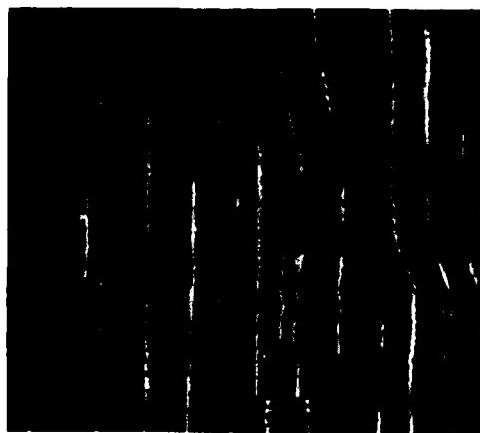


(e) Log magnitude

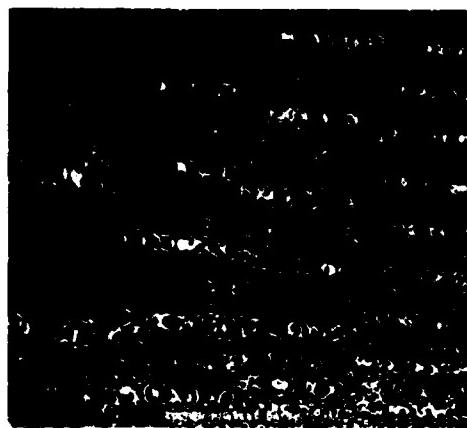


(f) Phase

signature number 4097-6144



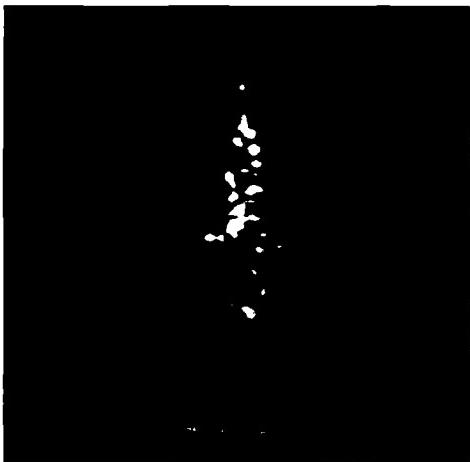
(g) Log magnitude



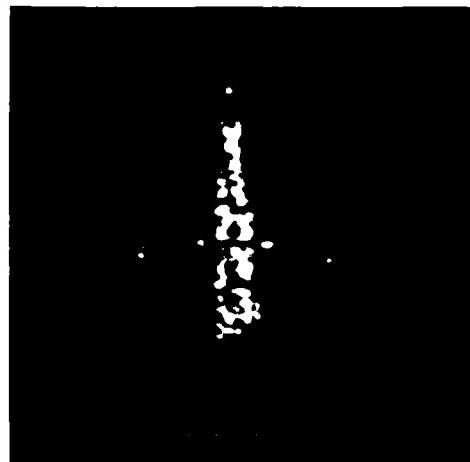
(h) Phase

signature number 6145-8000

Fig. 19. (continued).



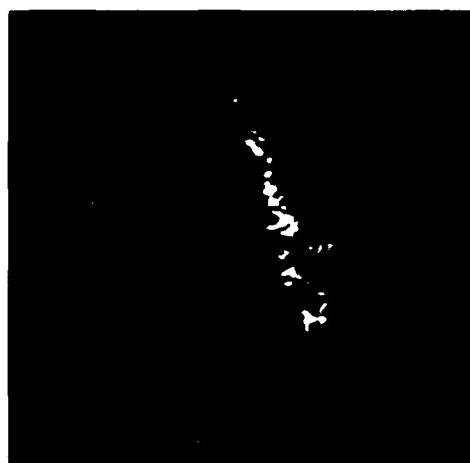
(a) 1st 2.5 seconds or 256 signatures ($\sim 4.5^\circ$ aspect change)



(b) 2nd 2.5 seconds

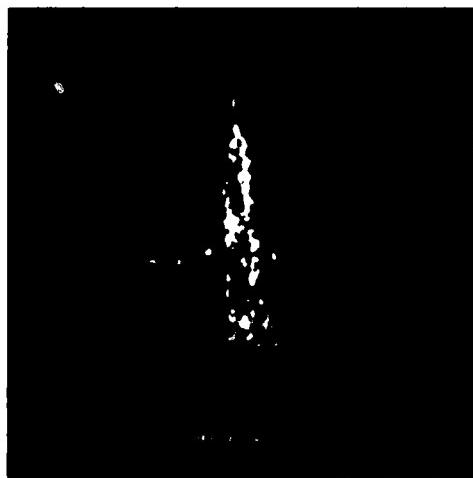


(c) 3rd 2.5 seconds



(d) 4th 2.5 seconds

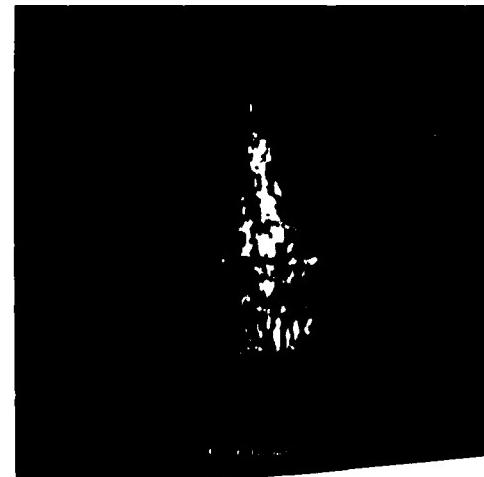
Fig. 20. Aircraft radar images with abutting 2.5 second coherence times.



(a) 1st 5 seconds
(~90° aspect change)

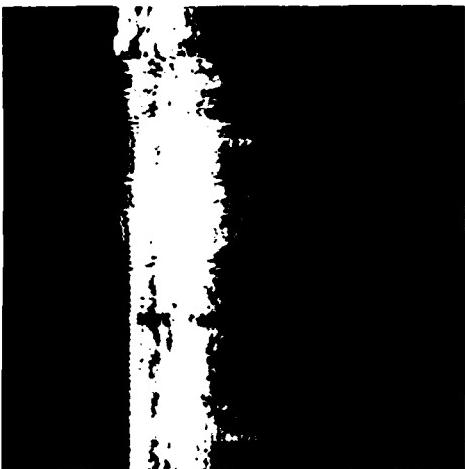


(b) 2nd 5 seconds

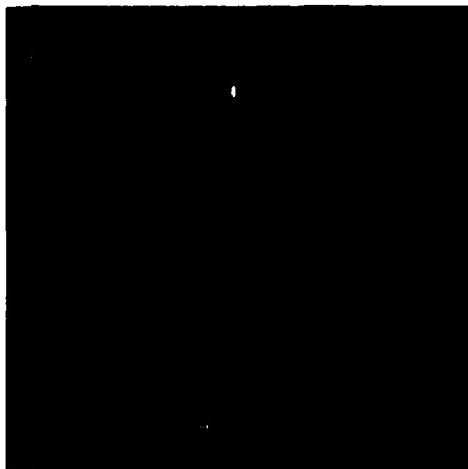


(c) 1st 10 seconds
(~18° aspect change)

Fig. 21. Aircraft images with different coherence times.



(a) Broken range lock



(b) Aircraft image before
re-alignment



(c) Correlation range re-
alignment



(d) Aircraft image after
re-alignment

Fig. 22. Range re-alignment.

processing. An earlier section presented a theoretical discussion on such re-alignment procedures and this section will present some experimental results.

Figure 22(a) presents a typical break in the range lock for a sequence of 512 signatures during the turning portion of the flight. The first returns, which are not very distinct in the first 50 and last 200 signatures, are from the nose tip. The second strong returns are from the left wingtip. Reflectivity of the nose tip scintillated and the wingtip returns were taken for the nose from time to time. Fig. 22(b) is the image of the data of Fig. 22(a). As one would expect, the image looks blurred due to the mixture of the returns from the wingtip and nose after the azimuth processing. However, general orientation of the fuselage is resolved.

A realignment scheme of correlating the magnitude of the returns as described in an earlier section was applied on Fig. 22(a) to become Fig. 22(c). While the scheme works quite well in the neighboring signatures, exponential weights have been applied to the previous aligned data for the correlation reference to insure global alignment.

Fig. 22(d) shows the target image obtained from the realigned data. Very much like Fig. 20 this image shows clearly the orientation and the wingtips of the aircraft. However greater structure is now evident as would be expected from properly realigned data.

References

1. Chen, C.C. and Andrews, H.C., "Multifrequency Imaging of Radar Turntable Data," submitted for publication in IEEE Transactions on Aerospace and Electronic Systems.

2. Harger, R.O., Synthetic Aperture Radar Systems, Academic Press, 1970.
3. Andrews, H.C. and Hunt, B.R. Digital Image Restoration, Prentice-Hall, 1977.
4. Leith, E.N., "Complex Spatial Filters for Image Deconvolution," Proceedings of the IEEE, Vol. 65, No. 1, January 1977.
5. Leith, E.N., "Range-Azimuth-Coupling Aberrations in Pulse Scanned Imaging Systems," JOSA, Vol. 63, No. 2, February 1973.

4. Smart Sensor Projects

The following report from Hughes Research Laboratories reflects the continuing progress on the CCD smart sensor design front. As usual we are pleased to see such results and wish to point out that this represents a classic illustration of technology transfer as the US Army NVL has contracted and received one of our earlier circuit chips in an operating unit. Recent chip design will afford 7x7 processing as well as programmable arrays and limited feature selection in our ultimate effort for the computation of a texture CCD circuit.

4.1 Charged Coupled Device Image Processing Circuitry

Graham N. Nudd

Program Overview

The goal of this project is to investigate the feasibility of performing low-level image-understanding tasks using charge-coupled device (CCD) technology. We have developed two CCD test chips which perform preprocessing functions based on a 3x3 array of pixels. The circuits developed perform the following functions:

- edge detection
- local averaging
- unsharp-masking
- adaptive stretch
- binarization.

The original data rate and accuracy goals were 100 kHz and six bits. To demonstrate these functions on as wide an image base as possible, we have developed a computer-based test facility that uses a dedicated 8-bit microprocessor. This system forms the interface between the USC IPI data base (stored on magnetic tape) and our integrated circuits. With this system we have been able to demonstrate each of the above functions at approximately 30 kHz with an overall accuracy of four bits. The speed and image resolution of this system is basically limited by the access time of the random access memory within the microcomputer. An access time of 1 μ sec provides for a 128x128 pixel image with 16 gray levels refreshed at standard frame rates. This combination of image size and resolution results in a processing rate of approximately 30,000 pixels/sec. The wide range of available images and the possibility of generating specific test pattern has permitted us to

demonstrate and evaluate the performance of our custom-built circuits on a wide selection of data.

The successful testing of these functions using the stored imagery has encouraged us to integrate them into a real-time system. The concept of this program has been to develop the necessary peripheral circuitry, to permit the CCD circuits developed to run directly from a commercial vidicon. The data rate from the Cohu camera is approximately 7.5 MHz equivalent to high-quality television. Although these circuits were not specifically developed to run at this data rate, the n-channel two-phase technology used in the circuit fabrication is capable of supporting this bandwidth. Our results to date with the real-time system indicate satisfactory operation at 3.9 MHz clock rates again with a dynamic range and accuracy equivalent to four bits. Using this system we have tested the processor on a variety of imagery.

In addition, we have started the initial concept development, circuit simulation, and preliminary layout on a third n-channel CCD chip. Some of the functions to be performed in this case (based on 7x7 array) include two-dimensional convolution with fixed bipolar weights for both deblurring and unsharp masking, a programmable 5x5 filter, a cross-shaped median operator and a statistical different operator. We encountered some delay in our original schedule for this chip, partly because of results of the simulation for the deblurring operation. However, we have developed a circuit concept for each of the functions (listed previously) and have begun the initial device layout for some of them.

As part of a parallel program with the Night-Vision Laboratories, Fort Belvoir, Virginia (Contract

DAAK70-77-C-0216) we have developed the necessary circuitry to operate the CCD circuits developed in this program as part of a real-time image preprocessing system. Details of this development, as appropriate to the operation of the circuits, are given in Section 4.

CCD Test Chips Developed

a. Test Chips I and II

We have developed five basic circuit functions, each operating on the 3x3 kernel shown in Figure 1. The functions are defined below Sobel Edge Detection:

$$S(e) = \frac{1}{8} [|(a + 2b + c) - (g + 2h + i)| + |(a + 2d + g) - (c + 2f + i)|] \quad (1)$$

Local averaging:

$$f_m(e) = \frac{1}{9} [a + b + c + d + e + f + g + h + i] \quad (2)$$

Unsharp masking:

$$S_u(e) = (1 - \alpha)e - \alpha f_m(e) \quad (3)$$

Adaptive binarization:

$$S_b(e) = \begin{cases} 1 & \text{for } f_m(e) \leq e \\ 0 & \text{for } f_m(e) > e \end{cases} \quad (4)$$

Adaptive stretch:

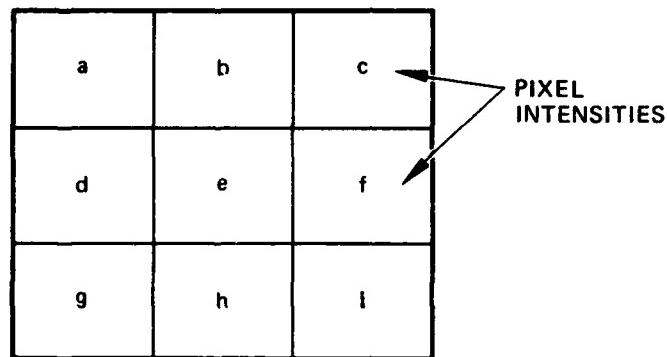


Figure 1. Schematic of the basic 3x3 kernel.

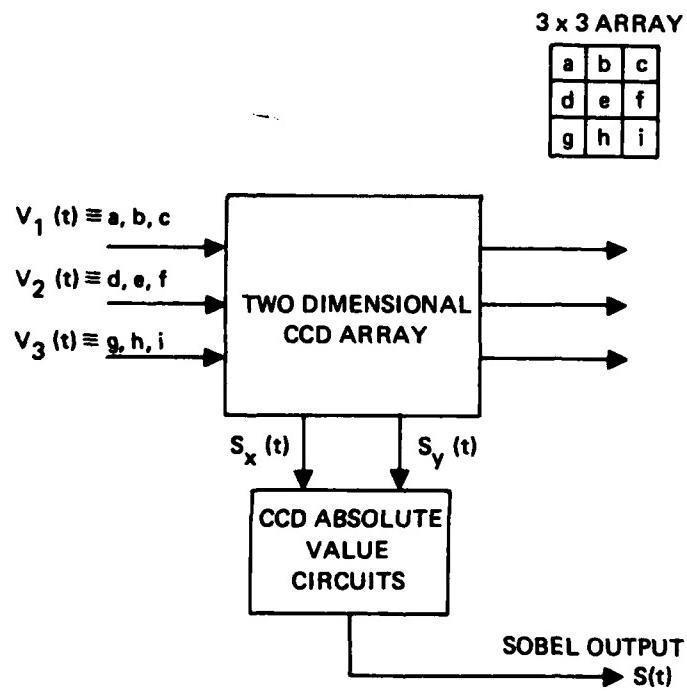


Figure 2. Block schematic of Sobel circuit.

$$S_a(e) = \begin{cases} 2 \text{ Min } \{e, r/2\} \\ 2 \text{ Max } \{(e, r/2), 0\} \text{ for } f_m(e) > r/2 \end{cases}$$

where r is the maximum pixel intensity. These circuits have been designed and built using two-phase n-channel technology with approximately $5\text{-}\mu\text{m}$ lithography. To achieve the required speed and accuracy, we have formulated each circuit as a two-dimensional transversal filter and an arithmetic operation. The concept of this approach is also shown in Figure 2 for the Sobel circuit. This approach allows all the advantages of the CCD transversal filtering functions developed over the past five years to be obtained and also provides one processed pixel output for each input data cycle. For the 3×3 processing discussed here, each filter must accept three adjacent lines of order and provide a processed output at each clock cycle as shown in Figure 3. The weighting necessary for each pixel ($1/8, 1/4, /8, -1/8, -1/4, -1/8$, etc.) is provided by variation in the area of floating gates.

The accuracy with which this weighting can be achieved depends to a large extent on the resolution of the lithography used. With standard optical alignment used in this work, an overall accuracy equivalent to 1% should be attainable. Details of the weighting necessary for both the Sobel operator and the local averaging are shown in Figure 4(a) and Figure 4(b), respectively.

The CCDs are n channel and are fabricated with a two-layer polysilicon process. This process requires nine masks and two ion implantations. The CCDs have a bit length of $27\text{ }\mu\text{m}$, and the minimum feature size is $2.5\text{ }\mu\text{m}$. This results in a total area of 0.7 mm^2 for the Sobel (see Figure 5(a)), of which 0.15 mm^2 is the transversal filter.

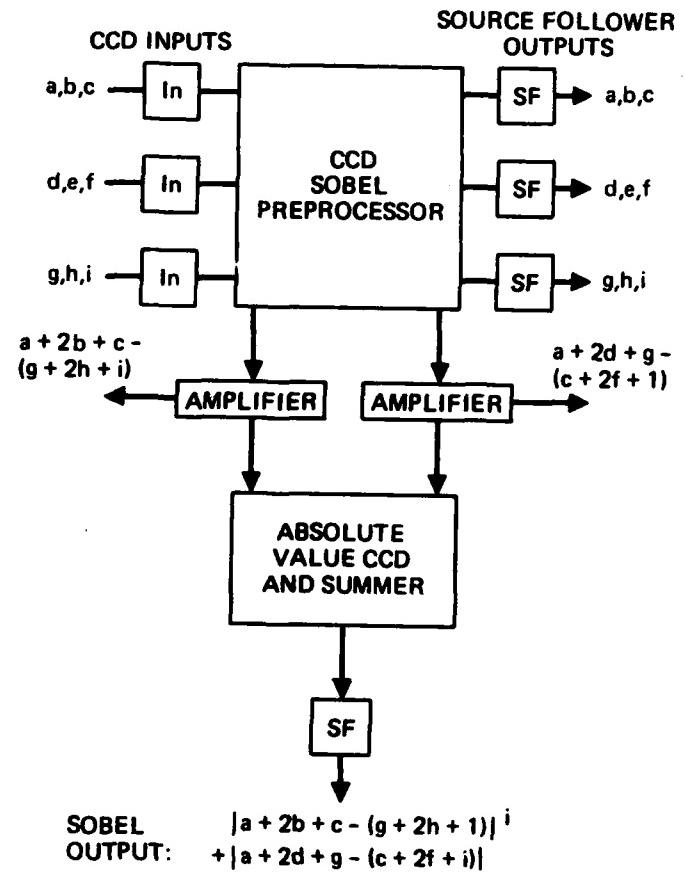


Figure 3. Detail of the CCD Sobel processor.

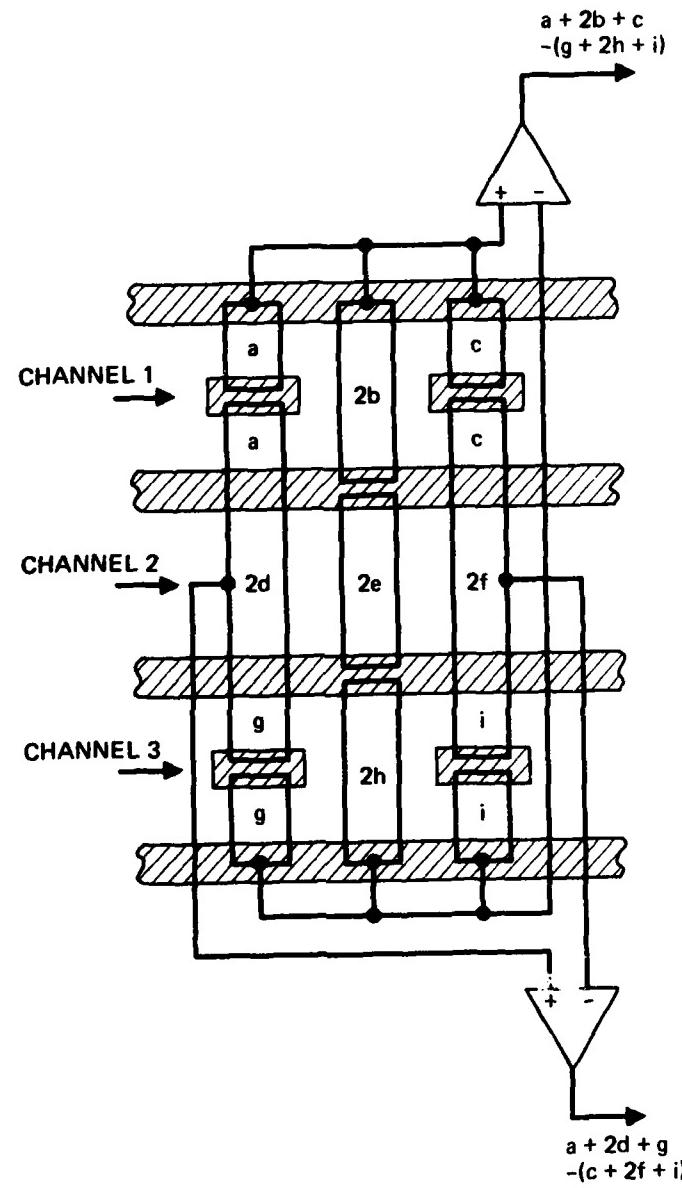


Figure 4(a). Schematic of the gate connection for the Sobel operator.

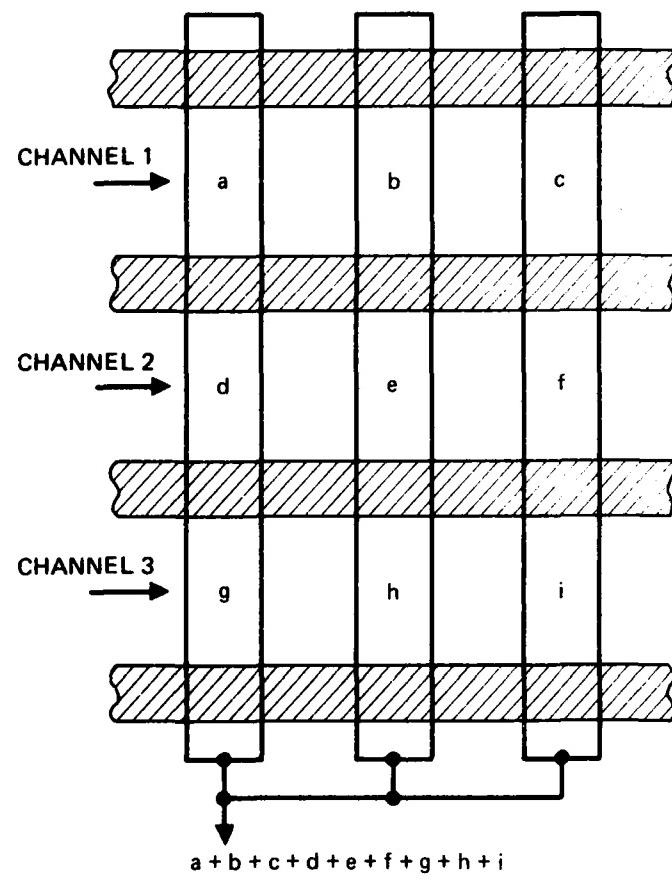
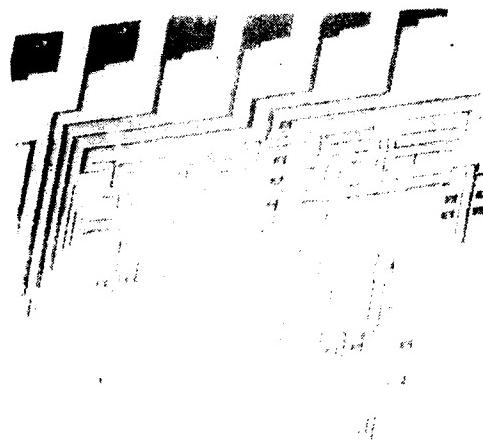
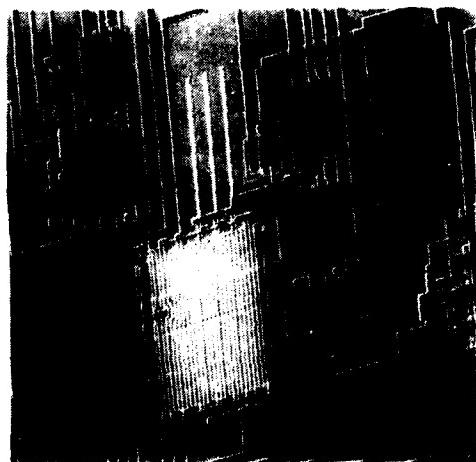


Figure 4 (b). Gate connection for the local averaging circuit.



(a)



(b)

Figure 5. Photomicrographs of (a) edge-detection and (b) the local mean filter.

This compares with a total area of 0.6 mm² for the mean filter (Figure 5(b)). To achieve the necessary capacitance balance between the two difference outputs, additional metal was added, as Figure 5 shows.

The two basic functions of the CCD circuits - arithmetic operations (such as absolute magnitude determination and summation) and transversal filtering - have been tested independently and the transfer characteristics have been measured. The weighting functions of the transversal filters for the Sobel edge detection and local mean evaluation, for example, can be written as:

$$S_x = 1/8 \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$S_y = 1/8 \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$W_m = 1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

where S_x and S_y provide the x and y components of the edge values, and W_m provides the mean. Both the impulse response and the linearity of these operations have been determined using the microcomputer-based test setup shown in Figure 6.

Here the microcomputer is used to provide flexible and programmed data inputs to the CCD circuits. These data are then clocked through the devices, and the output is stored

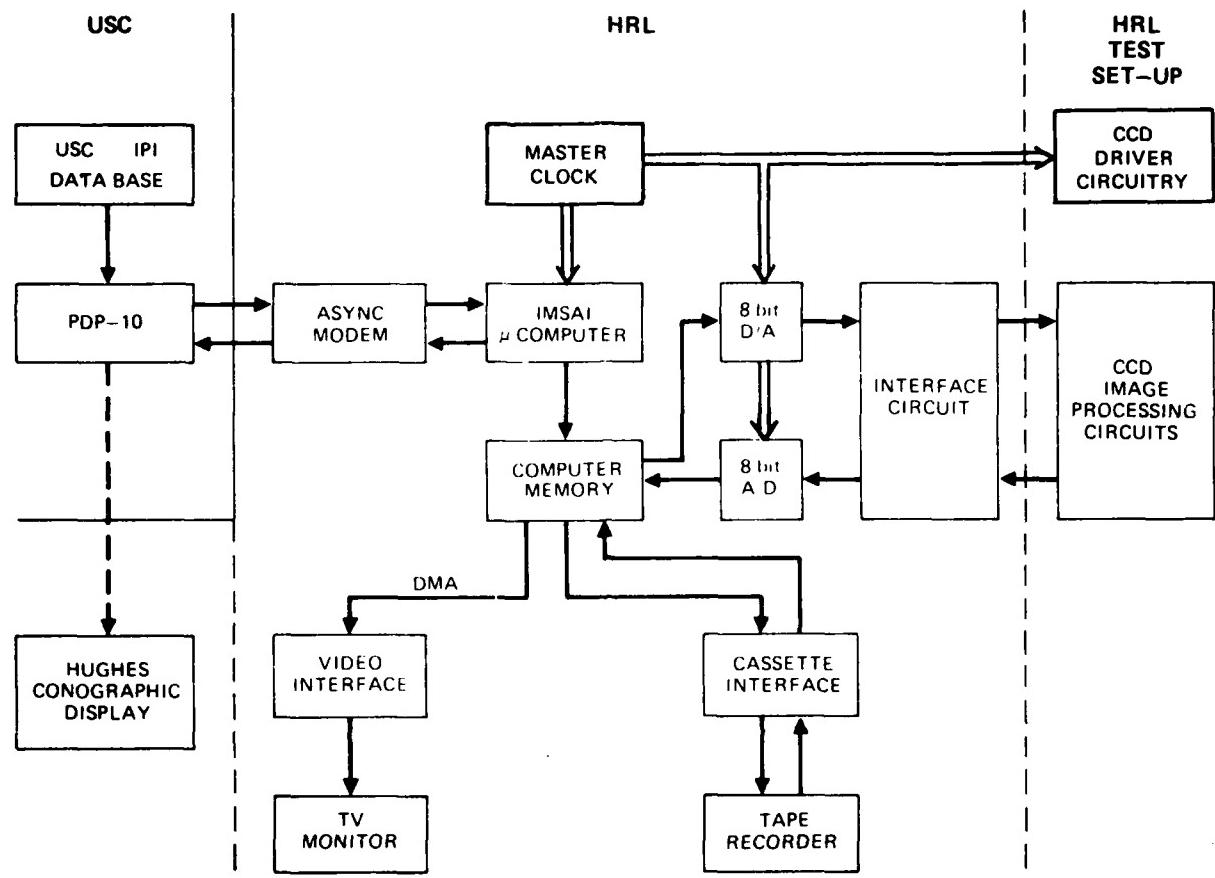


Figure 6. Microcomputer-based test facilities.

in the computer memory. This provides an accurate and rapid means of characterizing the device performance as a function of the various input parameters. The speed and accuracy of this system are basically determined by the computer cycle time and the analog-to-digital converters. The machine described here has a basic cycle time of $\approx 2 \mu\text{sec}$ and can provide an 8-bit quantization, resulting in a maximum CCD clock speed of $\approx 30 \text{ kHz}$.

When a single-input pulse with a duration of less than one-half clock cycle is used as the input, the output is equivalent to the impulse response of each component of the filters. Examples for the Sobel operation are shown in Figure 7.

An additional benefit of this test setup is that a unique pattern of either analog or digital data can be generated and used as the input to the CCD circuit, and the output data can be gated to uniquely determine the operation of any tap within the array. For example, if an input that linearly increases with time is clocked into the array and the output is gated to measure only the n th output pulse in each cycle, the weighting W_n of the n th floating electrode in the array can be uniquely determined. Measurements made in this way are shown in Figure 8, which shows the output voltage directly as a function of the input for each of the floating gates in the Sobel filter. The slope of each input/output characteristic gives the weighting for each tap. From this inputs can be shown to be linear over approximately a 3-V range, which translates to an accuracy and dynamic range equivalent to approximately 16 gray levels.

The CCD absolute value circuit shown in Figure 9, uses a novel technique which permits a charge storage that is

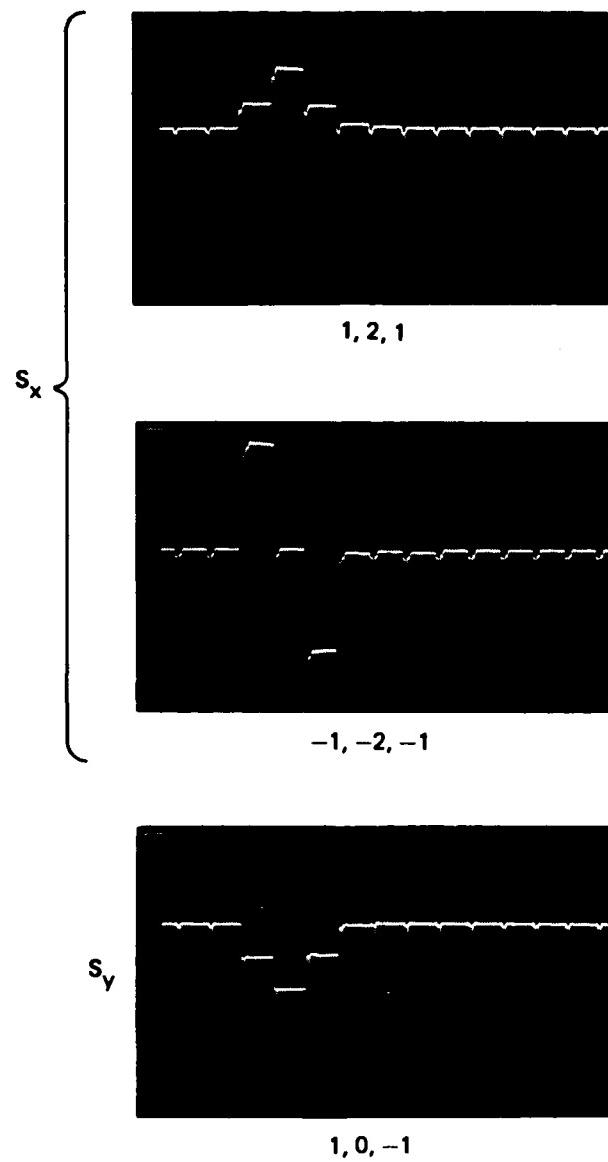


Figure 7. Measured impulse response of
CCD Sobel filter.

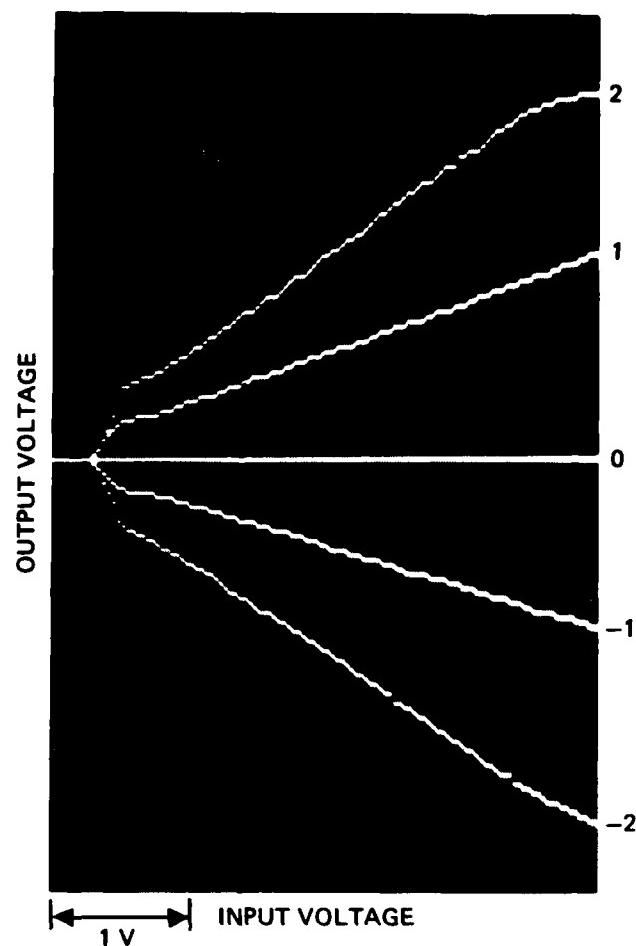
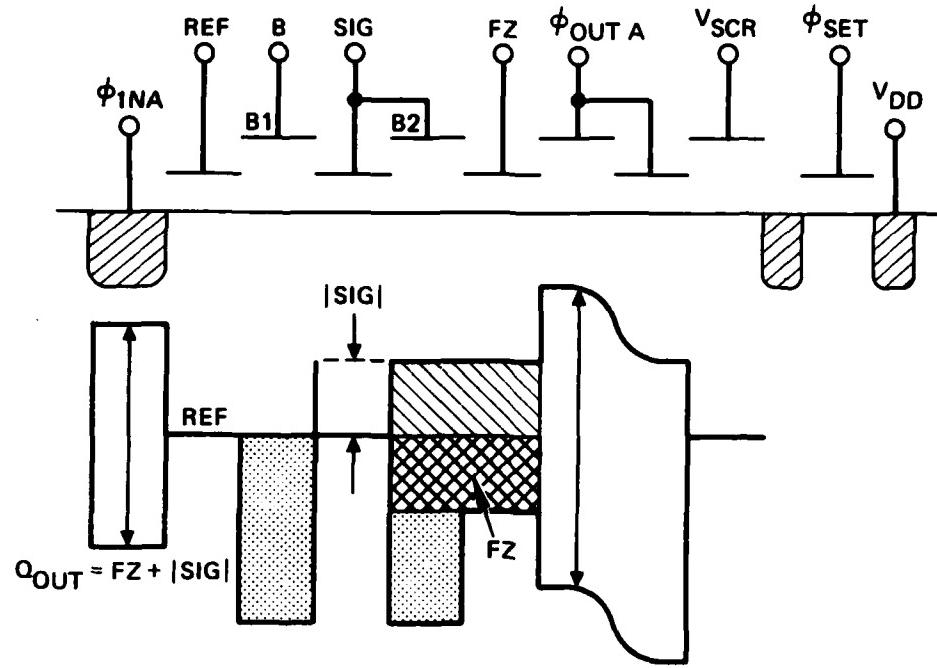
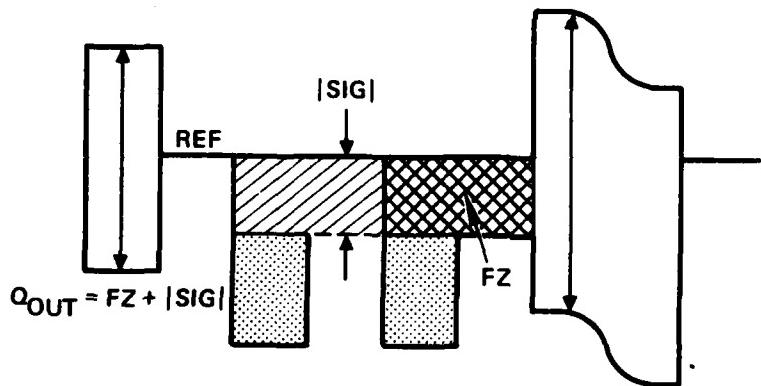


Figure 8. Measurement of the weighting functions for Sobel operators.



(a) $V_{SIG} < V_{REF}$



(b) $V_{SIG} > V_{REF}$

Figure 9. CCD absolute value circuit.

equivalent only to the input signal magnitude and is independent of signal polarity. During the input phase, ϕ_{INA} is pulsed low first (high surface electron potential in an n-channel CCD) and then settles high (low surface electron potential). When the signal voltage V_{SIG} is less than the reference voltage V_{REF} set by the REF gate, the electrons will fill the potential well under the gates B2 and FZ, as shown in Figure 9(a). During the output phase, ϕ_{OUTA} is pulse high, and the charge packet is transferred to the summing output. This charge is proportional to

$$[(V_{FZ} - V_{REF}) (A_{FZ} + A_{B2}) + (V_{REF} - V_{SIG}) (A_{FZ} + A_{B2})] ,$$

where A_{FZ} is the rate of the gate FZ, etc. The first term corresponds to the fat zero charge and the second to the signal charge. However, if V_{SIG} is higher than V_{REF} , the potential well under B1, SIG, B2, and FZ, will be filled, as shown in Figure 9(b). The output charge is proportional to

$$[(V_{FZ} - V_{REF}) (A_{FZ} + A_{B2}) + (V_{SIG} - V_{REF}) (A_{SIG} + A_{B1})] .$$

If the gate areas are fabricated so that $A_{SIG} + A_{N1} = A_{FZ} + A_{B2}$, then the output charge will always be a fat zero plus the charge proportional to the magnitude of the signal with V_{REF} as the reference point. A charge output corresponding to the absolute value of the input signal is obtained. After the absolute values of the differences are obtained, they are summed in the charge domain and the Sobel operation is completed.

Results of the absolute value circuit test are shown in Figure 10. The input voltage on the gate SIG has been swept over a range of 0 to 10 V.

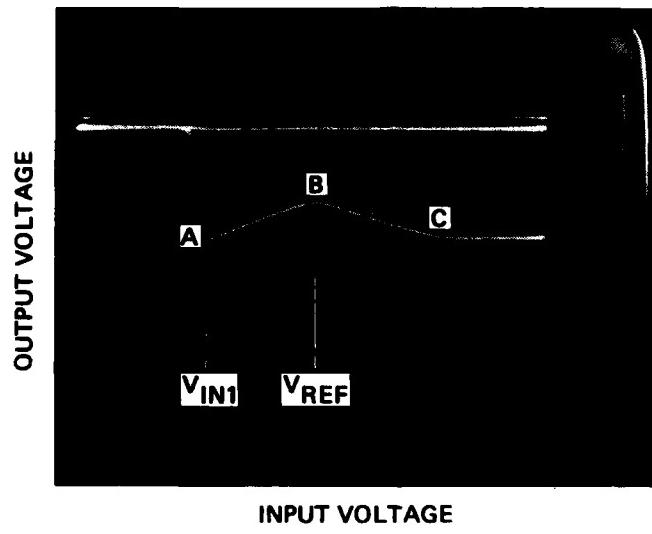


Figure 10. Transfer characteristic of absolute value circuit.

Initially, as the signal voltage is increased, charge flows over the input gate and is stored under gates FZ and B1. This charge is then clocked out as the clock phase changed. However, as the input voltage is increased beyond V_{INL} (Figure 10), the bucket size decreases linearly, resulting in the linear charge in voltage out (AB). When the input voltage reached V_{REF} , the bucket size is a minimum equivalent only to the fat zero. Increasing the input further, causes some of the charge previously trapped under B1 to be clocked out. Thus, the output characteristic again changes linearly from B to C. Consequently, when the input signal is operated about V_{REF} , the output changes linearly in proportion to $|V_{SIG} - V_{REF}|$ (when the output polarity is independent of V_{SIG}). The input voltage swing, as shown in Figure 10, is ≈ 2 V, resulting in an output change of some 400 mV (equivalent to an accuracy of ≈ 4 bits).

b. Performance Evaluation of the Processor

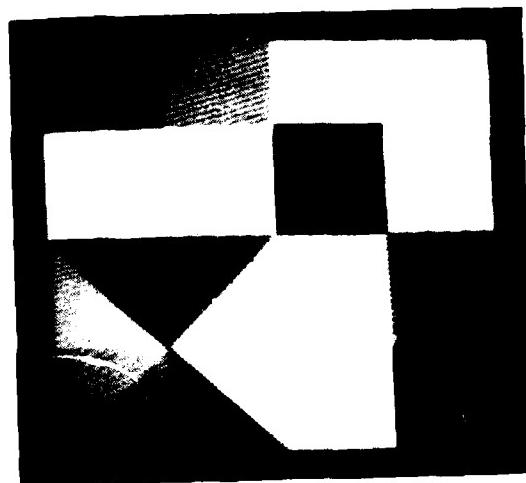
The processor has been tested on true two-dimensional imagery, using both a stored data base and a real-time input from a commercial vidicon. The use of a stored data base permits most of the problems associated with the sensor - illumination, resolution, and signal-to-noise ratio - to be separated from the evaluation of processor performance. The maximum data rate of this system, however, is limited to 30 kHz. In this mode, the imagery to be processed is first digitized and stored in the computer memory (as shown in Figure 6). In practice, a very large data base is available on magnetic tape and has been used extensively in the performance evaluation. The stored data are then clocked out of the random access memory in synchronism with the CCD clocks and converted to analog data before entering the processor. The processed data from the CCD are converted again to digital format and stored in the

computer memory in the form of 128x128 4-bit words. Direct memory address is then used to refresh a standard video monitor.

An example of this operation is shown in Figure 11 for a two-dimensional test pattern. A comparison of the output (Figure 11(b)) with the computer simulation (Figure 11(c)) shows that an accuracy of approximately four bits is preserved.

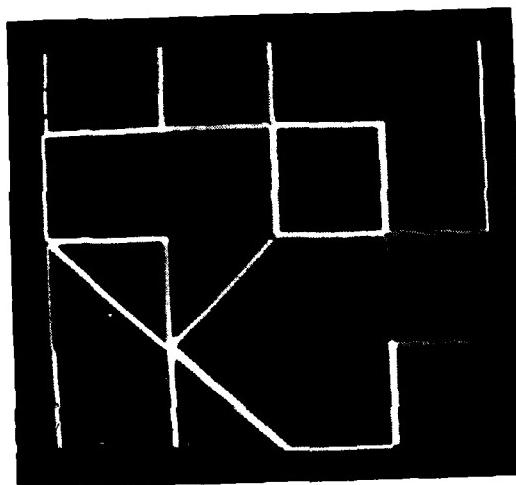
In addition to the performance evaluation of the CCD circuits on 128x128 pixel test images, we have tested their operation on a variety of imagery available both with the USC IPI data base and at Hughes Research Laboratories. Examples are shown in Figure 12 and 13. In addition to our work using the microcomputer-based test facility and stored imagery, we have interfaced the processor directly with a commercial vidicon camera. The standard operating frequency of this real-time video is \approx 7 MHz, providing 525x525 picture elements at 30 frames/sec. At present, we have operated our CCD processor at a maximum clock rate of 4 MHz, which provides the full 525 vertical resolution elements but about a 3:1 resolution loss in the horizontal direction. An example of the performance of the edge detection is given in Figure 14.

In parallel with this program, Night Vision Laboratories have funded a program to develop the interface circuitry for a real-time non-interlaced system which performs each of the operations defined in Eqs. 1 through 5. A schematic of the system is shown in Figure 15. It uses the devices developed in this program without modification and interfaces these with a Fairchild analog field delay as shown.



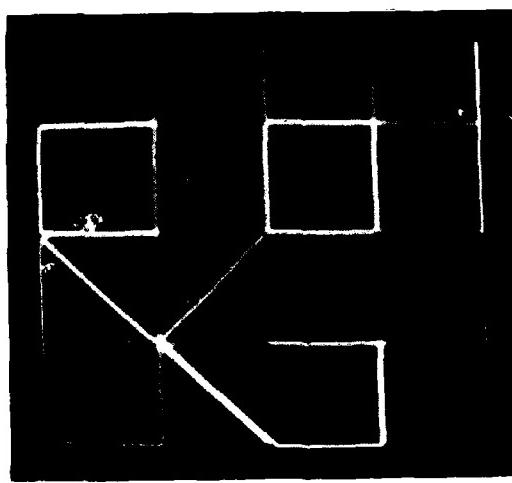
TEST 1

(a)



CHIP TEST NO. 120

(b)



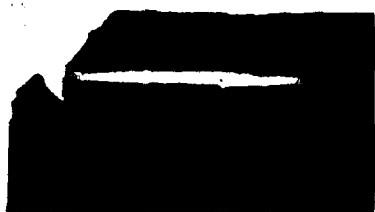
COMPUTER TEST 1

(c)

Figure 11. Example of processor operation on stored test data
(at 30 kHz and 128x128 pixel resolution).



ORIGINAL IMAGE

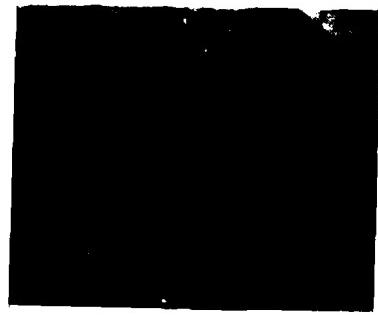


BLURRED IMAGE



SOBEL OF IMAGE

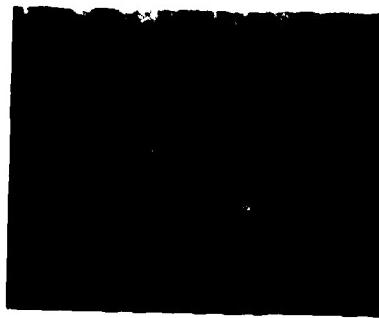
Figure 12. Example of processor operation
on stored imagery I.



ORIGINAL IMAGE

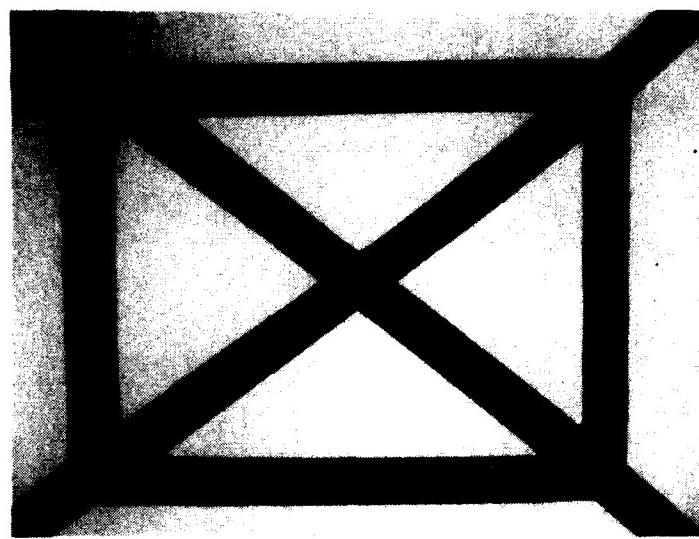


BLURRED IMAGE

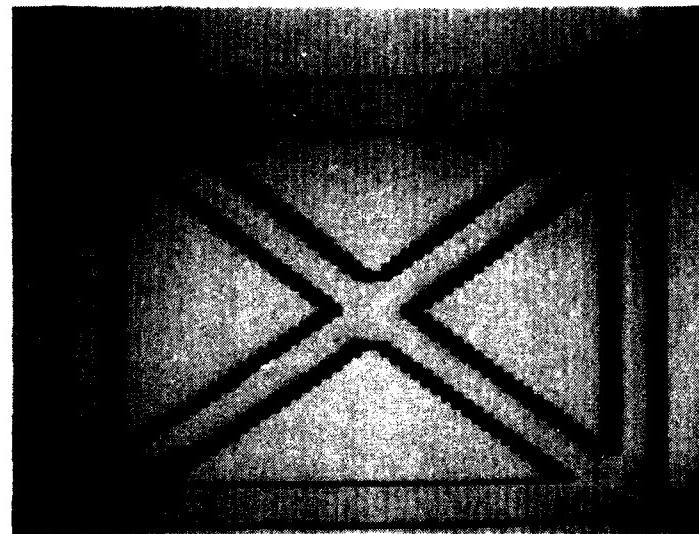


SOBEL OF IMAGE

Figure 13. Example of processor operation
on stored imagery II.



(a) ORIGINAL IMAGE



(b) EDGE DETECTED IMAGE

Figure 14. Example of the CED processor operation when clocked at 4 MHz using a commercial vidicon as the sensor.

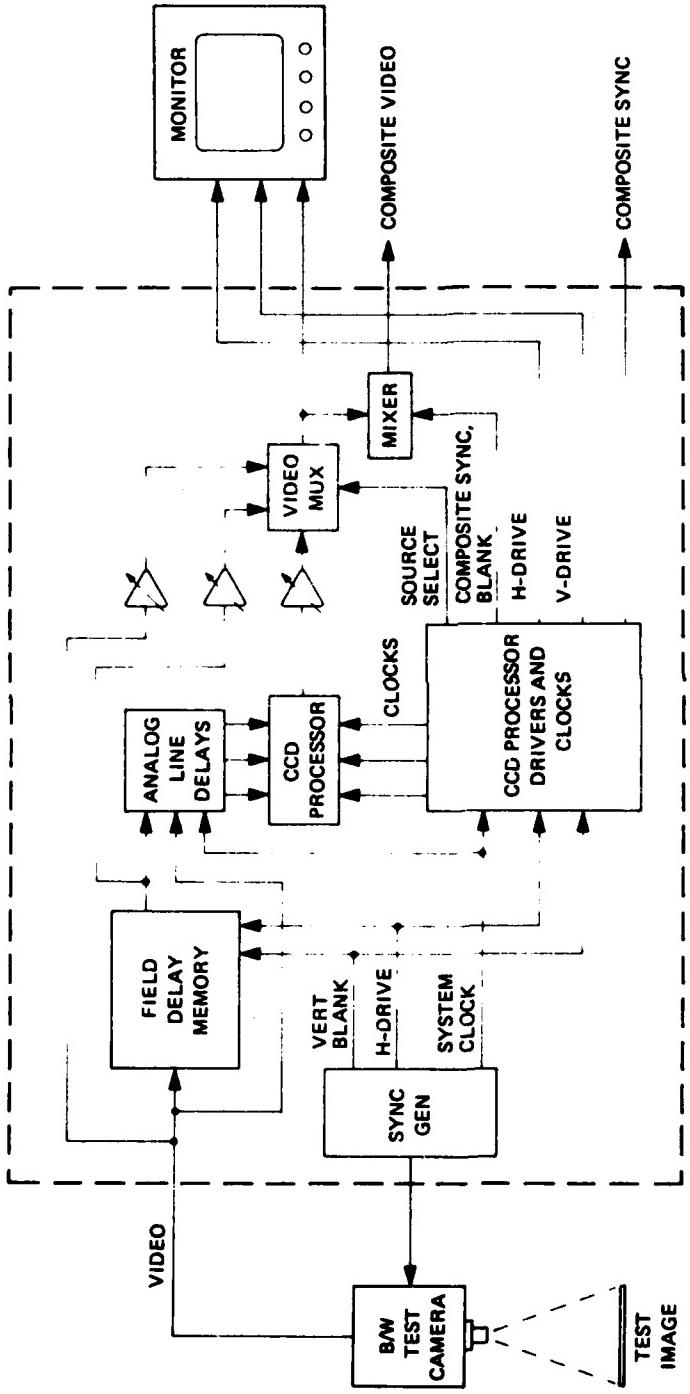


Figure 15. Schematic of real-time image processing system.

The results of this are shown in Figure 16, for local averaging, edge detection, unsharp masking and binarization. The overall performance of each operation depicted in Figure 16 is equivalent to four bits or 16 shades of gray.

We have begun work on a third test chip which is aimed at performing the following functions:

- 7x7 pixel bipolar linear filter
- Operator programmable bipolar linear filter
- Data programmable linear filter
- Texture related 7x7 parameter development.

The goal of the program is to develop circuitry that will run at the full television rates (≈ 7 MHz) and provide at least 6-bit (1 part in 64) levels of intensity revolution.

The 7x7 pixel bipolar array suggested by USC IPI has the weighting elements shown below:

$$\begin{bmatrix} -0.05 & -0.03 & 0.04 & 0.06 & 0.04 & -0.03 & -0.05 \\ -0.03 & 0.06 & -0.02 & -0.07 & -0.02 & 0.06 & -0.03 \\ 0.04 & -0.02 & -0.06 & 0.15 & -0.06 & -0.02 & 0.04 \\ 0.06 & -0.07 & 0.15 & 0.68 & 0.15 & -0.07 & 0.06 \\ 0.04 & -0.02 & -0.06 & 0.15 & -0.06 & -0.02 & 0.04 \\ -0.03 & 0.06 & -0.02 & -0.07 & -0.02 & 0.06 & -0.03 \\ -0.05 & -0.03 & 0.04 & 0.06 & 0.04 & -0.03 & -0.05 \end{bmatrix}$$

Its purpose is to perform correction for a limited aperture size, and essentially de-blurr the imagery. Originally, the weights suggested were given to four significant places which we feel is impossible to achieve with photolithographic techniques. Essentially, the accuracy achievable in a CCD filter will be limited by the minimum

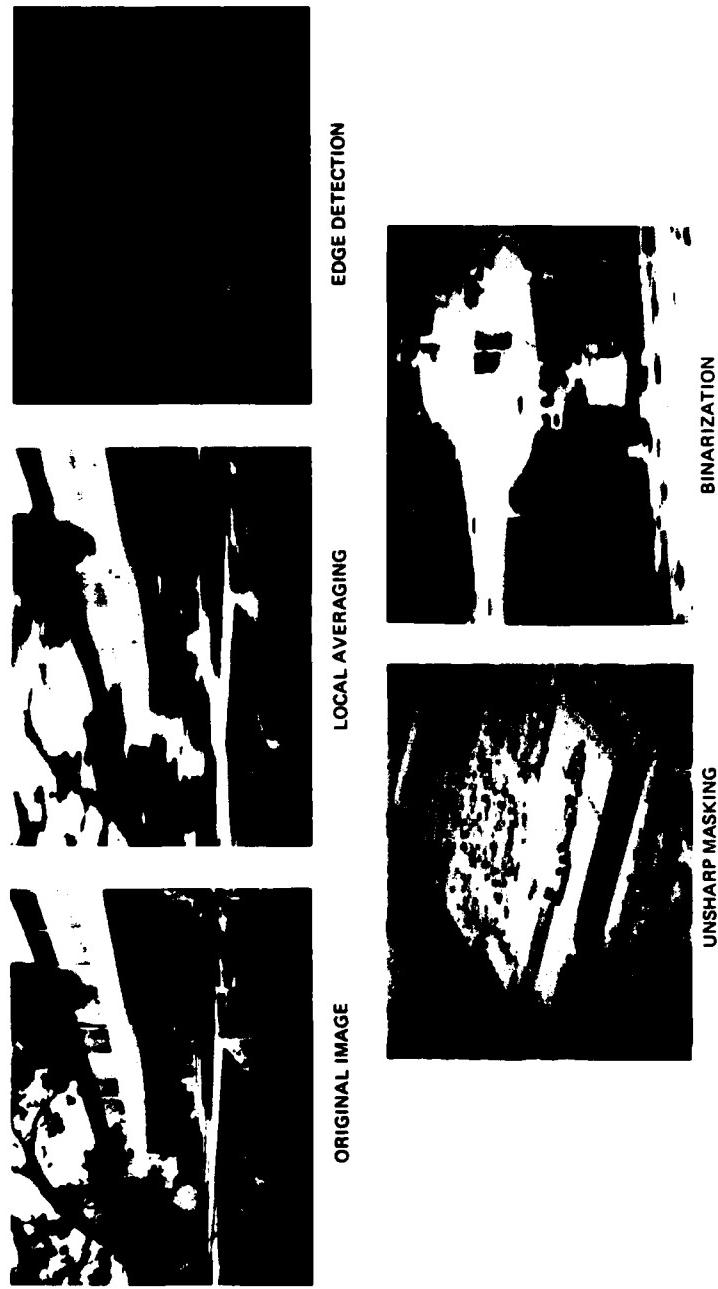


Figure 16. Performance of processor at real-time data rates.

feature size in proportion to the length of the channel width as shown in Figure 17. The weighting at each tap is simply given by W_i/W , where W_i is the length of the floating gate and W is the width of the channel. Typically, the minimum feature size achievable after full processing using photolithography, will be on the order of 2 μm resulting in an overall accuracy of about one percent.

We have simulated this function at HRL, and the performance improvement using the weights shown is slight. This issue has been raised with the USC group and they intend to rework the design. Because of the expense involved in the detailed design layout and processing of the circuit and the testing, we do not intend to build this junction until improved performance can be demonstrated.

The second array is a programmable bipolar filter with the weightings shown below,

$$\begin{bmatrix} -0.039\alpha & -0.0078\alpha & -0.01172\alpha & -0.0136\alpha & -0.0117\alpha & -0.0078\alpha & -0.039\alpha \\ -0.0078\alpha & -0.0156\alpha & -0.0234\alpha & -0.0312\alpha & -0.0234\alpha & -0.0156\alpha & -0.0078\alpha \\ -0.0117\alpha & -0.0234\alpha & -0.0312\alpha & -0.0469\alpha & -0.0312\alpha & -0.0234\alpha & -0.0117\alpha \\ -0.0136\alpha & -0.0312\alpha & -0.0469\alpha & (1 - 0.0625\alpha) & -0.0469\alpha & -0.0312\alpha & -0.0156\alpha \\ -0.0117\alpha & -0.0234\alpha & -0.0312\alpha & -0.0469\alpha & -0.0312\alpha & -0.0234\alpha & -0.117\alpha \\ -0.0078\alpha & -0.0156\alpha & -0.0234\alpha & -0.0312\alpha & -0.0234\alpha & -0.0136\alpha & 0.0078\alpha \\ -0.039\alpha & -0.0078\alpha & -0.0117\alpha & -0.0156\alpha & -0.0117\alpha & -0.0078\alpha & -0.039\alpha \end{bmatrix}$$

This array is now being designed and laid out. We will be implementing this operation as a unipolar array and access to the center pixel. A schematic of this is given in Figure 18.

The third circuit is a median operator working on a 5x5 (plus-shaped) array. We aim to provide at least four-bit

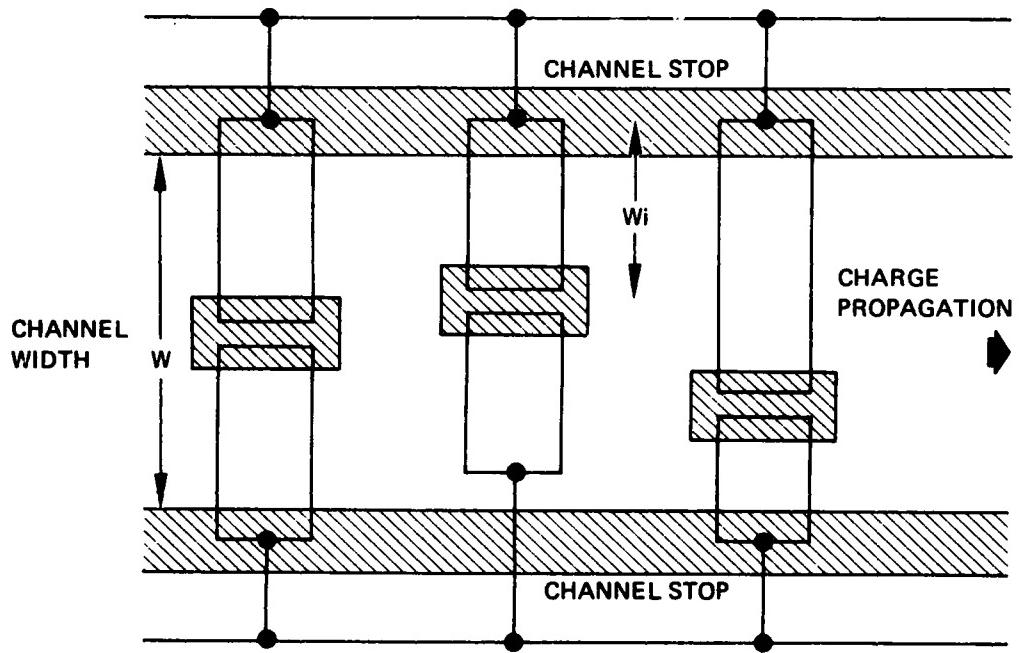


Figure 17. Illustration of floating gate weighting technique.

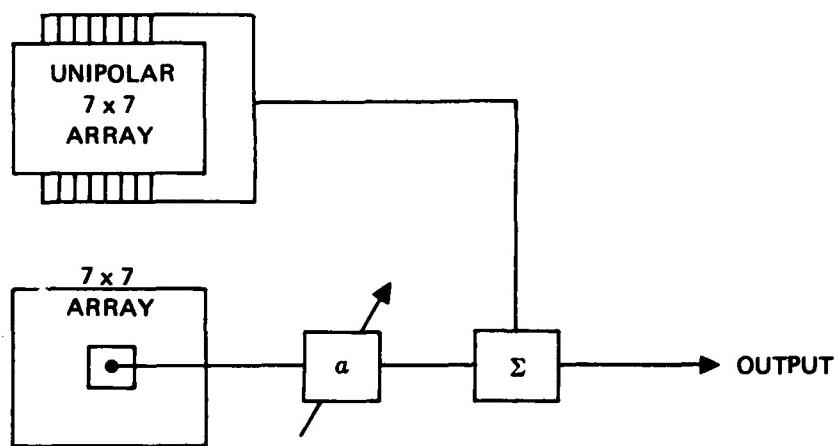


Figure 18. Schematic of data programmable array.

accuracy with a goal of n_x bits and provide a complete calculated output each video clock cycle. This typically involves ranking the ten pixels in the array and then performing a further ten comparisons. We are attempting to organize this as a two-dimensional filtering operation and limit its automatic operations to a single high-speed parallel comparator operation.

In addition, we are investigating the possibility of performing a histogram on a 7×7 array with four-bit accuracy directly in the analog domain. This circuit could probably perform a complete histogram in 50 to 100 μsec . We will use this to perform standard statistical measurements, such as standard deviations, mode filtering, and dispersion calculations.

In addition, we are now designing and beginning to lay out a 5×5 programmable filter with programmable weighting junctions. It will operate at the real-time video rates, and the processed data will be fed to a microprocessor that will change its weights. This will provide the high-speed operation required for the data processing and a capability of changing the filter junction with about a 1% accuracy at the frame rate.

Finally, we are working on a 3×3 Laplacian operator that operates at real-time video rates.

Each of these circuits will be built using standard photolithographic techniques and using n-channel two-phase operations. We expect initial devices to be available by year end. To support this testing phase of this program, we are building a test setup that will provide the necessary access to seven adjacent lines of data and have programmable clock rates.

Contributors to this program include: Scott Fouse,
Graham Nudd, Paul Hygaard, and Gary Thurmond.

5. Recent Ph.D. Dissertations

One of the Image Processing Institutes' most precious products is its graduate students and it is always a pleasure to see our students graduate and move on to professional positions. This section lists the abstracts of the dissertations of the three most recent graduates and represents research in edge detection, restoration, and radar imaging. We are proud of their work and wish them well in their endeavors. Details of their dissertations appear as USCIPI technical reports and are available upon request for those interested.

5.1 Quantitative Methods of Edge Detection

Ikram E. Abdou

Most local operators used in edge detection can be modelled by one of two models: edge enhancement/thresholding and edge fitting. This dissertation presents a quantitative design and performance evaluation of these methods. The design techniques are based on statistical detection theory and deterministic pattern recognition classification procedure. The performance evaluation methods developed include: (a) deterministic measurement of the edge gradient amplitude; (b) comparison of the probabilities of correct and false edge detection; and (c) figure of merit computation. The design techniques developed are used to optimally design a variety of small and large mask edge enhancement/thresholding operators. A performance comparison is given between these edge detectors. A new edge fitting algorithm is introduced. The new algorithm is derived in the discrete domain, this allows a direct optimization of the operator's performance. The advantages of the new algorithm are better performance with real world pictures and less sensitivity to signal-to-noise ratio.

5.2 An Investigation Into an A Posteriori Method of Image Restoration

John B. Morton

Two algorithms are developed which address the problem of estimating the magnitude and phase of the optical transfer function associated with a blurred image. The primary focus of the research is on the estimate of the phase of the optical transfer function. With the sharpening

AD-A134 989

IMAGE UNDERSTANDING RESEARCH(U) UNIVERSITY OF SOUTHERN
CALIFORNIA LOS ANGELES IMAGE PROCESSING INST
H C ANDREWS 30 SEP 78 USCIP-840 F33615-76-C-1203

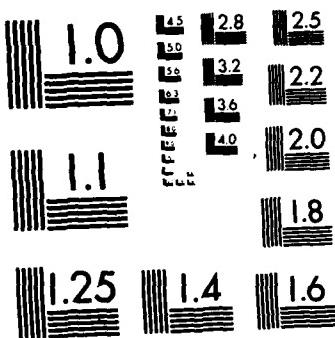
4/4

UNCLASSIFIED

F/G 20/6

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

of one approximation, the method affords a reasonable estimate of the phase of the optical transfer function. Once an estimate of the optical transfer function has been made, the corresponding blurred image is Wiener filtered to estimate the original unblurred image. Results are demonstrated on computer simulated blurs and also on real world blurred imagery. Included is a mathematical bound on the phase of the optical transfer function.

5.3 Imaging With Radar Returns

Chung-Ching Chen

This dissertation presents both analytic and processing techniques for various radar imaging systems.

A two dimensional system classification method, which is very general and hence applies to the special case of radar imaging systems as well, is proposed to assist in understanding the structure and describing the limitations of 2-D systems. Once a given system is identified with the simplest possible class, the specific techniques can be directly utilized to process the data or reconstruct the images.

Following a review of radar imaging principles, several coherent radar systems are analyzed and experimented upon. They include synthetic aperture radar (SAR) ground mapping, imaging of an aircraft target from turntable data, and imaging of a flying aircraft target. In each case the point spread function (PSF) of the imaging system is derived or estimated. Physical considerations are then incorporated in mathematical PSF's to categorize the imaging systems according to the aforementioned system classification

principle proposed. Degrees of Freedom (DOF) under different imaging geometries are analyzed as a means to determine the amount of information present in the usually huge amount of raw radar data for the purpose of efficient computation and minimal storage requirement. Motion compensation, range curvature, range alignment, de-chirping, FFT, registration and side lobe reduction problems are all addressed and experiments are performed using data from RAT-SCAT (for turntable imaging) and other facilities. The results shown suggest the versatility of coherent radar imaging.

Possible extentions of the current work are discussed. The understanding of the system characteristics, in particular the formation of the radar image will aid in the advancement of techniques for radar image enhancement, encoding, quantization, and restoration.

6. Recent Institute Personnel Publications

1. H.C. Andrews, (ed) Tutorial and Selected Readings in Digital Image Processing, IEEE Computers, Long Beach, Ca. 1978.
2. J. Bescos and T.C. Strand, "Optical Pseudocolor Encoding of Spatial Frequency Information," Applied Optics, Vol. 17, 2524 (August 15, 1978).
3. C.C. Chen and H.C. Andrews, "Data Structures of Two Variable to Two Variable Mappings," submitted to IEEE Transactions on Circuits and Systems.
4. C.C. Chen and H.C. Andrews, "Multifrequency Imaging of Radar Turntable Data," submitted to IEEE Transactions on Aerospace and Electronics Systems.
5. C.C. Chen and H.C. Andrews, "Target Motion Induced Radar Imaging," submitted to IEEE Transactions on Aerospace and Electronics Systems.
6. H.S. Hou and H.C. Andrews, "Cubic Splines for Image Interpolation and Digital Filtering," accepted for publication, IEEE Transactions on Acoustics, Speech and Signal Processing (Dec 1978).
7. Y.S. Hsu, S. Prum, J.H. Kagel, and H.C. Andrews, "Pattern Recognition in the Mandala Cosine Domain," submitted to IEEE Transactions on Computers.
8. D.G. McCaughey and H.C. Andrews, "The Continuous-Discrete Model: Least Squares Inverses and Singular Function Expansion," submitted to IEEE Transactions on Information Theory.

9. D.G. McCaughey and H.C. Andrews, "Variable Knot Splines for Image Approximation," accepted for publication, IEEE Transactions on Computers.
10. J.B. Morton and H.C. Andrews, "An A Posteriori Method of Image Restoration," submitted to JOSA.
11. J.B. Morton and H.C. Andrews, "Computing Phases in A Posteriori Image Restoration," submitted to IEEE Transactions on Acoustics, Speech and Signal Processing .
12. F. Naderi and A.A. Sawchuk, "Detection of Low Contrast Images in Film-Grain Noise," Applied Optics, Vol. 17, (September 15, 1978).
13. F. Naderi and A.A. Sawchuk, "Estimation of Images Degraded by Film- Grain Noise," Applied Optics, Vol. 17, pp. 1228-1237, (April 15, 1978).
14. R. Nevatia and K. Price, "Locating Structures in Aerial Images," to appear in the Proceedings of the Third International Joint Conference on Pattern Recognition, Kyoto, Japan, November 1978.
15. W.K. Pratt and O.D. Faugeras, "Development and Evaluation of Stochastic- Based Visual Texture Features," Fourth International Joint Conference on Pattern Recognition, Kyoto, Japan, November 1978.
16. W.K. Pratt, O.D. Faugeras, and A. Gagalowicz, "Visual Discrimination of Stochastic Texture Fields," IEEE Transactions on Systems, Man and Cybernetics, November 1978.
17. K. Price, "Symbolic Matching and Analysis with

Substantial Changes in Orientation," Proceedings of ARPA Image Understanding Workshop, Cambridge, Mass. May 1978.

18. A.A. Sawchuk, "Artificial Stereo," Applied Optics, Vol. 17, (December 15, 1978).

19. A.A. Sawchuk and C-K. Hsueh, "Computer Generated Double Phase Holograms," Applied Optics, Vol. 17, (December 15, 1978).

20. A.A. Sawchuk, A. Armand D. Boswell, B.H. Soffer, and T.C. Strand, "Real-Time Nonlinear Optical Processing with Liquid Crystal Devices," Proceedings IEEE 1978 International Optical Computing Conference, London, September 1978.

21. A.A. Sawchuk, A. Armand, D. Boswell, B.H. Soffer, and T.C. Strand, "Approaches to Nonlinear Optical Processing in Real Time," Proceedings International Commission for Optics Congress, Madrid, Spain, September 1978.

22. A.A. Sawchuk and T.C. Strand, "Space-Variant Processing with Polychromatic Light," Proceedings International Commission for Optics Congress, Madrid, Spain, September 1978.

23. A.A. Sawchuk, A. Armand, D. Boswell, J. Michaelson, B.H. Soffer and T.C. Strand, "Real-Time Nonlinear Processing with Halftone Screens," 1978 Annual Meeting, Optical Society of America, San Francisco, October 1978, Journal Optical Society of America, Vol. 68, (October 1978).

24. A.A. Sawchuk, A. Armand, D. Boswell, B.H. Soffer and T.C. Strand, "New Methods for Real-Time Nonlinear

Optical Processing," 1978 Annual Meeting, Optical Society of America, San Francisco, October 1978, Journal Optical Society of America, VOL. 68, (October 1978).

25. A.A. Sawchuk, L.M. Frantz, and W. Von der Ohe, "Real-Time Optical Phase Measurement," 1978 Annual Meeting, Optical Society of America, San Francisco, October 1978, Journal Optical Society of America, Vol. 18 (October 1978).
26. T.C. Strand and H. Bescos, "Hybrid Optical-Electronic Processing Applied to Chest Radiographs," Journal of Applied Photographic Engineering, Vol. 4, (Fall 1978).

END

FILMED

12/83

DRUG